



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Grabación, posproducción de video 360 y desarrollo de una
plataforma de streaming adaptativo y reproducción del
mismo incluyendo efectos multi-sensoriales

Trabajo Fin de Grado

Grado en Ingeniería de Sistemas de Telecomunicación, Sonido e
Imagen

AUTOR/A: Egea Todolí, Erica

Tutor/a: Boronat Segui, Fernando

CURSO ACADÉMICO: 2022/2023

Resumen

Este TFG está basado en el desarrollo y la integración de efectos multi sensoriales (aromas) en vídeos 360°. Este proceso se realizará mediante *software* generador de metadatos sobre los efectos y dispositivos encargados de generar dichos efectos. Además, también se incluye la gestión del contenido en un servidor web para poder realizar *streaming* del mismo por redes IP.

El contenido 360° generado será preparado para utilizar *streaming* adaptativo basado en HTTP y la técnica *tile-based streaming*. Con el *streaming* adaptativo se consigue que la calidad del vídeo varíe según las condiciones de red o características del dispositivo del usuario, consiguiendo que el vídeo se reproduzca en la mayor calidad posible en todo momento. La técnica *tile-based streaming* permite transmitir solo la región de la imagen que está siendo visualizada, obteniendo un ahorro considerable de ancho de banda.

Para ello se requiere un trabajo previo de grabación y posproducción del material. Para ello, se grabarán y editarán los vídeos 360° siguiendo un plan de rodaje y un guion técnico previamente realizados, dotando así al producto final de un desarrollo narrativo. Asimismo, se incluye la creación de una página web con visor de vídeo 360° compatible con todo lo anteriormente mencionado.

Palabras clave

Video 360°, mulsemmedia, posproducción, HAS, *tile-based streaming*

Abstract

This TFG is based on the development and integration of multi-sensory effects (scents) into 360° videos. This process will be carried out through metadata-generating software for the effects and devices responsible for generating these effects. Additionally, it also includes content management on a web server to enable streaming over IP networks.

The generated 360° content will be prepared for use with Adaptive Streaming over HTTP and the tile-based streaming technique. Adaptive streaming ensures that the video quality varies based on network conditions or user device characteristics, always ensuring that the video plays in the highest possible quality. The tile-based streaming technique allows for transmitting only the region of the image being viewed, resulting in significant bandwidth savings.

To achieve this, there is a need for prior recording and post-production work on the material. For this purpose, 360° videos will be recorded and edited following a previously prepared shooting plan and technical script, thereby giving the final product a narrative development. Additionally, it includes the creation of a web page with a 360° video viewer compatible with all the elements.

Keywords

360° video, mulsemmedia, post-production, HAS, *tile-based streaming*

Tabla de contenido

1. Introducción.....	7
1.1 Objetivos	8
1.2 Metodología	8
1.3 Estructura de la memoria.....	9
1.4 Dificultades del proyecto.....	10
2. Los vídeos 360°.....	10
2.1 El auge de los vídeos 360°: desde la era digital a la actualidad	10
2.1.1 El aspecto inmersivo	10
2.1.2 Desde la era digital hasta la actualidad	11
2.2 Vídeos 360° o realidad virtual. Diferencias y similitudes	13
2.3 La narrativa de los videos 360°	14
2.3.1 El emplazamiento de cámara	16
2.3.2 La iluminación	16
2.4 Los vídeos 360° en el sector turístico.....	18
3. Codificación en <i>tiles</i>	19
3.1. ¿Qué es?	19
3.2 La técnica <i>tile coding</i> en los videos 360°	20
3.2.1 El problema de la distorsión.....	21
4. <i>Streaming</i> Adaptativo basado en HTTP (HAS)	25
4.1 Tecnologías disponibles.....	25
4.1.1 <i>Streaming</i> con descarga progresiva tradicional.....	26
4.1.2 <i>Streaming</i> con descarga progresiva basada en HTTP.....	27
4.1.3 <i>Streaming</i> adaptativo basado en HTTP (HAS).....	27
4.2 Sistemas basados en tecnologías HAS (<i>Streaming</i> Adaptativo basado en HTTP)	28
4.2.1 Comparación de los cuatro sistemas HAS	33
4.3 MPEG-DASH.....	34
4.3.1 El fichero índice y sus componentes	34
5. Sistemas mulsemmedia	36
5.1 Entornos multisensoriales	36
5.1.1 Construcción de un entorno mulsemmedia	37
5.2 Estandarización. MPEG-V.....	38
5.3 Estructura del SEM.....	39
6. Grabación del material	42
6.1 Proceso de preproducción.....	42
6.1.1 Localizaciones del proyecto.....	42
6.1.2 Guion técnico	42

6.1.3 Plan de rodaje	46
6.1.4 Cámara utilizada. Características y <i>software</i>	46
6.1.1 Cámara utilizada	46
6.1.2 Software	48
6.2 Grabación del material	49
6.3 Exportación del material	49
7. Proceso de posproducción	52
7.1 Organización del material	52
7.1.1 Visionado del material	52
7.2 Edición del material	53
7.2.1 Animación inicial	53
7.2.2 Recorridos en bicicleta	54
7.2.3 Edición de videos 360°	57
7.3 Tratamiento del sonido	59
7.4 Corrección de color	60
7.5 Exportación	62
8. Proceso de codificación	63
8.1 Generación de los vídeos en distintas calidades	63
8.2 División en <i>tiles</i>	65
8.2.1 Activación del sistema WSL (Windows Subsystem for Linux)	66
8.2.2 Instalación del codificador Kvazaar	66
8.2.3 Creación de <i>tiles</i>	68
8.3 Generación de segmentos DASH y su mpd	69
8.3.1 Empaquetar el flujo de bits	69
8.3.2 Generación segmentos DASH y su mpd	69
8.4 Creación del servidor HTTP y almacenamiento de los segmentos	70
8.5 Reproducción del contenido	71
9. Inclusión de efectos multisensoriales	72
9.1 Instalación <i>software</i> SEVino2	72
9.2 Creación de efectos muntisensoriales	76
10. Conclusiones	79
Bibliografía	81

ÍNDICE DE TABLAS

Tabla 1 Resumen diferencias entre realidad virtual y video 360°.....	14
Tabla 2 Resumen de las características de las distintas proyecciones.....	25
Tabla 3 Resumen de las características de las tecnologías de transmisión	28
Tabla 4 Resumen HLS.....	29
Tabla 5 Resumen MSS.....	32
Tabla 6 Resumen DASH.....	33
Tabla 7 Resumen características HDS, HLS, MSS.....	33
Tabla 8 Relación resoluciones y bitrate en función de la calidad.....	64
Tabla 9 Explicación de cada uno de los parámetros.....	65
Tabla 10 Explicación parámetros del proceso de creación de tiles.....	69
Tabla 11 Explicación parámetros generación del archivo .mpd.....	70

ÍNDICE DE ILUSTRACIONES

Ilustración 1 Gráfico continuo virtualidad – Realidad.....	11
Ilustración 2 Headsight de Philco.....	12
Ilustración 3 Sensorama (HMD de heiling).....	12
Ilustración 4 Diferencia 3 grados de libertad frente 6 grados de libertad	14
Ilustración 5 Codificación por tiles.	19
Ilustración 6 Ejemplo del FoV en la codificación por tiles.	21
Ilustración 7 Proyección equirrectangular.....	23
Ilustración 8 Diagrama de la equal-area cylindrical projection.....	23
Ilustración 9 Diagrama de la Cubic Mapping	24
Ilustración 10 Estructura fragmento Smooth Streaming.....	31
Ilustración 11 Estructura Smooth Streaming.....	31
Ilustración 12 Niveles jerárquicos archivo mpd	35
Ilustración 13 Máscara Feelreal	37
Ilustración 14 Diseño genérico de un sistema mulsemmedia.....	38
Ilustración 15 Estructura declaración	40
Ilustración 16 Estructura grupo de efectos.....	40
Ilustración 17 Ejemplo grupo de efectos.....	40
Ilustración 18 Estructura efecto.....	41
Ilustración 19 Ejemplo de efecto	41
Ilustración 20 Estructura efecto de definición	42
Ilustración 21 Idea de cómo realizar los recorridos entre ubicaciones	43
Ilustración 22 Las cuatro localizaciones ubicadas en un mapa	44
Ilustración 23 Guion técnico.....	44
Ilustración 24 Guion técnico “Parpalló”	45
Ilustración 25 Guion técnico “Marjal”	45
Ilustración 26 Guion técnico “Alqueria”	45
Ilustración 27 Guion técnico “Playa”	45
Ilustración 28 Plan de rodaje	46
Ilustración 29. Especificaciones de la cámara Qoocam 8k Enterprise	47
Ilustración 30. Especificaciones de la cámara en la retransmisión en vivo	47
Ilustración 31 Tutoriales app QooCam App.....	48
Ilustración 32. Organización de los brutos	49
Ilustración 33. Preview del vídeo sin el software QooCam Studio 2.0.....	50
Ilustración 34. Pantalla de inicio del QooCam Studio 2.0.	50
Ilustración 35 Ajustes en el software QooCam Studio.	51
Ilustración 36 Menú de opciones de Stitch.	51
Ilustración 37 Primer nivel de organización del material.	52
Ilustración 38 Segundo nivel de organización del material.....	52
Ilustración 39 Resumen edición con Google Earth Studio.	54
Ilustración 40. Mapa recorrido Parpalló-Marjal	55
Ilustración 41. Mapa recorrido Marjal – Alqueria	55
Ilustración 42. Mapa recorrido Alqueria-Playa	55
Ilustración 43 Bicicleta.....	56
Ilustración 44 Diferencia de opacidad entre el recorrido ya realizado y el que está por realizar.....	56
Ilustración 45. Organización material recorridos	57
Ilustración 46. Ejemplo organización del material en Adobe Premier Pro 2023.....	58
Ilustración 47. Organización de los recorridos en Adobe Premier Pro 2023.	58
Ilustración 48 Edición de la música	60

Ilustración 49 Ejemplo transición entre un clip de la canción y el siguienteE	60
Ilustración 50 Timeline ordenado	61
Ilustración 51 Sistema de zonas de Ansel Adams.....	61
Ilustración 52 Ejemplo gráfica forma de onda.	62
Ilustración 53 Ajustes de exportación.....	63
Ilustración 54 Consola PowerShell de Windows	66
Ilustración 55 Instrucciones para clonar repositorio de GitHub.	67
Ilustración 56 Menú de opciones para clonar el repositorio desde la app Github para Windows.....	67
Ilustración 57 Organización de los segmentos en función de la calidad. Ejemplo localización playa	71
Ilustración 58 Ejemplo segmentos almacenados en la subcarpeta 1080.....	71
Ilustración 59 Menú de opciones para clonar el repositorio	71
Ilustración 60 Versión de Java a descargar	72
Ilustración 61 Acceder a las variables del sistema (Paso 1)	73
Ilustración 62 Acceder a las variables del sistema (Paso 2)	74
Ilustración 63 Ejemplo variables	74
Ilustración 64 Ejemplo variables path.....	75
Ilustración 65 Ejemplo archivo SEVino2.bat.	75
Ilustración 66 Iniciar software SEVino2.	76
Ilustración 67 Ventana de inicio aplicación SEVino2.	76
Ilustración 68 Controles de reproducción y volumen.....	77
Ilustración 69 Información del video	77
Ilustración 70 Timeline y shot-timeline	77
Ilustración 71 Pista de efectos.....	78
Ilustración 72 Ajustes de efectos.....	78
Ilustración 73 Ejemplo archivo SEM generado con SEVino2.....	79

1. Introducción

Desde que aparecieron los primeros medios de comunicación hasta hoy en día, ha habido un factor común a lo largo de la historia de los medios audiovisuales, y no es otro que la constante evolución y el desarrollo de nuevas tecnologías. Dentro de esta evolución, siempre se ha asociado el vídeo como el formato clave en campos como la información, la publicidad o el ocio. Dentro de este formato, cabe destacar que actualmente existe un auge de los vídeos verticales como forma de entretenimiento a través de dispositivos móviles, y de los videos 360° como forma de tener una experiencia mucho más inmersiva. De hecho, un estudio de Media Brix (una plataforma de publicidad) demostró que los anuncios en videos verticales tienen una tasa de finalización un 90% mayor en comparación con los vídeos horizontales debido a que la audiencia se engancha durante más tiempo a un vídeo vertical que a uno horizontal (Marketing audiovisual, 2022).

En los últimos años, ha tenido lugar un notable crecimiento la realización y el uso de vídeos 360°, en cierta medida gracias a los avances tecnológicos y también debido a que los usuarios cada vez prestan más interés en obtener una experiencia más enriquecedora y gratificante. Este auge de los videos 360° se ha notado, sobre todo, en el sector turístico y comercial, pues hay personas que, por distancia, falta de tiempo o falta de recursos económicos, no pueden disfrutar de ciertos lugares de forma presencial, y gracias a la tecnología y los videos 360°, lo pueden hacer desde su casa.

Este aumento actual, tanto del uso de los vídeos 360° como del interés de los usuarios de dejar de ver para “sentir” cada vez más el contenido que están consumiendo es lo que ha dado pie a llevar a cabo este proyecto. Este proyecto es un Trabajo Final de Grado (TFG) conjunto de los grados de Ingeniería de Sistemas de Telecomunicación, Sonido e Imagen (GISTSI) y Comunicación Audiovisual (GCAU). Por lo tanto, el proyecto no solo cuenta con los procesos de grabación y posproducción de videos 360° de algunos de los lugares icónicos de Gandía y sus alrededores (la Cueva del Parpalló, la Marjal, la Alquería del Duc y la playa de Gandía, concretamente en el paseo marítimo de esta), sino que también incluye el desarrollo de una plataforma de *streaming* adaptativo y de reproducción del contenido generado, incluyendo efectos multisensoriales (aromas) a través de un difusor de aromas profesional, teniendo como objetivo el enriquecimiento de la experiencia del usuario

Por lo tanto, con el objetivo de mejorar la experiencia del usuario en todo momento, se utiliza el *streaming* adaptativo basado en HTTP o HAS (*HTTP-based Adaptive Streaming*) y la técnica *tile-based streaming*. Con el *streaming* adaptativo basado en HAS se consigue que la calidad del vídeo pueda variar en función de la calidad de la conexión del usuario, lo que hace que el vídeo se reproduzca siempre en la mayor calidad posible y, además, evita que el usuario pierda información del contenido o tenga parones en la reproducción de este. Por otro lado, la técnica *tile-based streaming* consiste en dividir la imagen en distintas regiones llamadas *tiles*, lo que permite transmitir solo la región de la imagen que está siendo visualizada por el usuario en ese momento en concreto, consiguiendo un considerable ahorro en el ancho de banda.

1.1 Objetivos

En cuanto a lo referente a los objetivos de este TFG, se pueden englobar en un objetivo principal que se puede dividir en varios objetivos secundarios. El objetivo principal del TFG consiste en realizar la grabación y posproducción de contenidos de video 360° sobre lugares icónicos de Gandía y desarrollar una plataforma basada en web de *streaming* adaptativo y reproducción de dichos contenidos, incluyendo efectos multi-sensoriales (aromas).

Mientras que los objetivos secundarios son los siguientes:

- Compaginar lo aprendido en ambos grados en un solo producto final.
- Realizar la grabación y posproducción de los vídeos de forma de que el producto final cuente con un desarrollo narrativo.
- Aprender a utilizar *streaming* adaptativo basado en HAS (*HTTP-based Adaptive Streaming*).
- Aplicar correctamente la técnica *tile-based streaming* en videos 360°.
- Gestionar el contenido en un servidor web.
- Crear y gestionar una página web que permita la reproducción del contenido y se comunique con el difusor de aromas, permitiendo la inclusión de estos.

1.2 Metodología

En primer lugar, se ha analizado la transición desde la era digital hasta la actualidad. También se ha llevado a cabo un estudio sobre la inmersión de la realidad virtual o RV, analizando la diferencia entre esta y los videos 360°. Además, se ha estudiado la

narrativa de los videos 360° y se han abordado los aspectos técnicos como el emplazamiento de cámara y la iluminación de dichos videos 360°.

En segundo lugar, se han estudiado todos los aspectos relacionados con la codificación en *tiles* y el *streaming* adaptativo basado en HTTP (HAS). A continuación, se ha realizado la grabación y edición del material seleccionado. Luego, se ha abordado el aspecto de la creación de los *tiles* y la preparación del contenido para utilizar el *streaming* adaptativo basado en HAS y en *tiles* (*tile-based streaming*).

Seguidamente, se han generado los metadatos sobre los efectos multisensoriales a incluir en cada uno de los vídeos y se ha modificado un reproductor web *open-source* para poder generar dichos efectos en los instantes adecuados de la reproducción de los videos 360° generados. Además, se ha desarrollado un código basado en NodeJS para poder interactuar con el dispositivo comercial encargado de la generación de los aromas.

Finalmente, se ha desarrollado, a modo de demostración, una página web incluyendo al reproductor en la que se permite seleccionar uno de los vídeos generados de una lista de reproducción, y reproducirlo con los aromas apropiados.

Por último, cabe destacar que durante el proyecto se han consultado diversas fuentes bibliográficas, para las cuales se ha utilizado el sistema de citación APA 7 (Normas APA, 2019)

1.3 Estructura de la memoria

A lo largo del Capítulo 2 se realiza un estudio de los videos 360°, haciendo referencia a su aspecto inmersivo y su auge en los últimos años, además de comentar el aspecto narrativo de estos. En cuanto a la codificación en *tiles*, en el Capítulo 3, se explica en qué consiste y cómo se utiliza en los videos 360°. Se hace especial énfasis en los problemas de distorsión y las distintas proyecciones que existen. A continuación, en el Capítulo 4, se explican las distintas tecnologías a la hora de hacer *streaming*, como el *streaming* con descarga completa, el *streaming* con descarga progresiva y el *streaming* adaptativo. Se finaliza esta sección explicando en detalle el *streaming* basado en HTTP y los cuatro sistemas basados en HAS que existes actualmente : *HTTP Live Streaming* de Apple, *MS Smooth Streaming* de Microsoft, *HTTP Dynamic Streaming* de Adobe y *Dynamic Adaptive Streaming over HTTP* de MPEG.

El Capítulo 5 hace referencia a los sistemas mulsemmedia, explicando en qué consisten y cuál es su estándar. A continuación, en los Capítulos 6 y 7 se abordan los procesos de preproducción, grabación y posproducción del material. En estos puntos se comentan las localizaciones escogidas, el plan de rodaje seguido y el material técnico utilizado, entre otros aspectos.

Finalmente, en los Capítulos 8 y 9 se presenta la parte práctica en cuanto al uso de la técnica *tile-based coding* y de las técnicas HAS, así como el proceso de inclusión de efectos multisensoriales. La memoria finaliza en el Capítulo 10 con las conclusiones del TFG llevado a cabo.

1.4 Dificultades del proyecto

Durante la realización de este proyecto se han encontrado diversas dificultades, la mayoría de las cuales son debidas a que, en su mayoría, las herramientas utilizadas para el trabajo no habían sido usadas con anterioridad. Del mismo modo, este proyecto ha sido la primera toma de contacto con ciertos softwares utilizados durante el transcurso de este, como por ejemplo SEVino2 o Adobe After Effects, por lo que para poder utilizarlos se ha llevado a cabo un proceso de aprendizaje de estos.

Por otro lado, en cuanto a las técnicas *tile-based coding* y *streaming*, y las técnicas HAS, se ha encontrado que los tutoriales están hechos para ser utilizados en el sistema operativo Linux. Por lo que para poder trabajar con un dispositivo con Windows se ha tenido que trabajar a través del sistema *Windows Subsystem for Linux* (WSL), teniendo como consecuencia la aparición de errores debidos a la incompatibilidad de ciertos procesos o archivos.

2. Los vídeos 360°

2.1 El auge de los vídeos 360°: desde la era digital a la actualidad

2.1.1 El aspecto inmersivo

Los vídeos 360° ofrecen una vista desde todas las perspectivas, a diferencia de los vídeos tradicionales, que solo permiten la vista bidimensional, por lo que tener una vista

panorámica es la característica clave de los videos 360° (Rytewiki, 2021). Esta inclusión de toda la escena implica una inmersión en esta, lo que se traduce en un acercamiento entre dicha escena y el espectador. Pese a que los videos 360° empezaron a surgir y tener su auge hace apenas una década, la intención de incluir, e incluso sumergir, al espectador en la historia es un aspecto que se ha considerado desde el inicio de la comunicación. La inmersión es la estrategia para remover los límites y la distancia psicológica entre el observador y el espacio de la imagen (Grau, 1999).

A principios de la década de los 90, Paul Milgram y Fumio Kishino plantean un gráfico llamado “Continuo Virtualidad-Realidad”, representando en él una unión entre el mundo real y la realidad virtual (ver ilustración 1). Entre ambos extremos se encuentran la realidad aumentada y la virtualidad aumentada.

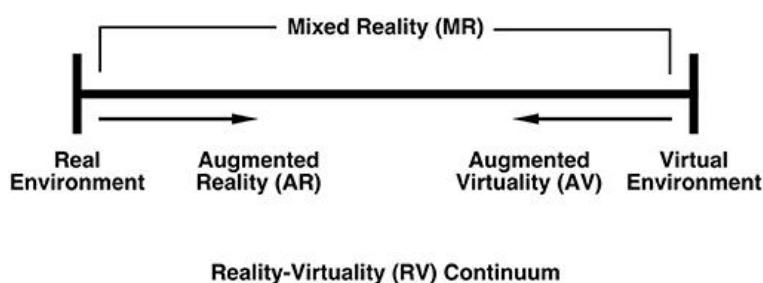


Ilustración 1 Gráfico continuo virtualidad – Realidad. Fuente: <https://www.frontiersin.org/articles/10.3389/frvir.2021.647997/full>

Dentro del bloque de la realidad virtual se encuentra tanto el vídeo 360° como el vídeo inmersivo, siendo el primero el foco de estudio de este TFG. La diferencia entre realidad virtual y un vídeo 360° se explica detenidamente en el apartado 2.2. No obstante, se basa, principalmente, en que, para considerarse realidad virtual, debe ser inmersivo e interactivo, mientras que los videos 360° no son interactivos, ya que en ellos no se permite interactuar con el entorno (Nielsen & Laura, 2022).

2.1.2 Desde la era digital hasta la actualidad

Pese a que anteriormente al término “realidad virtual” existen precedentes directos como pueden ser el *Headsight* de Philco, el HMD (*Head Mounted Displays*) de Heiling o el casco VIVED de Fisher (ver ilustraciones 2 y 3), no es hasta 1985 cuando se empieza a usar la realidad virtual en la industria del entretenimiento y no solo en los laboratorios. Este salto se da de la mano de Jaron Lanier, quien funda VPL (*Visual Programming Language*) y desarrolla cascos y guantes de realidad virtual, además de distintos softwares comerciales. El propio Lanier es el creador del término “realidad virtual”. Este salto y desarrollo tecnológico por parte de Lanier, es lo que permite catalogar la realidad

virtual como “nuevo medio” , es decir, objetos no solo exhibidos y distribuidos por ordenador, sino también producidos digitalmente (Manovich, 2005,p.64).

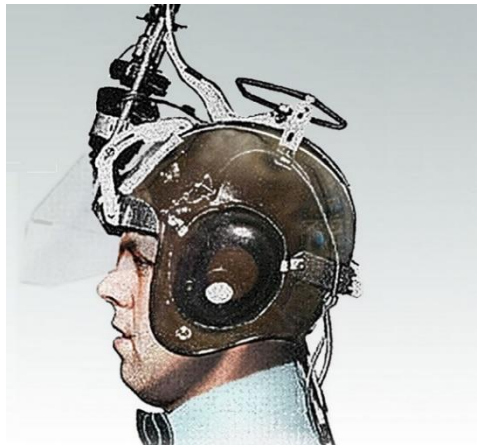


Ilustración 2 Headsight de Philco. Fuente: <https://www.virtual-reality-shop.co.uk/philco-headsight-1961/>



Ilustración 3 Sensorama (HMD de heiling). Fuente: <https://www.netzpiloten.de/sensorama-virtual-reality-in-1962/>

A partir de este momento, tanto grandes compañías como diversas universidades empiezan a estudiar la realidad virtual y a apostar por ella, siendo esta el principal objeto de interés y, por tanto, protagonista de libros, revistas, conferencias o incluso películas, como es el caso de Matrix (Wachowski, 1999). Sin embargo, era tal el crecimiento de estudio y desarrollo de la realidad virtual que la tecnología de ese entonces no era capaz de afrontar dichos avances, por lo que a finales de la década de los 90 y principios del 2000, se paraliza la industria de la realidad virtual, conociéndose este periodo como el “invierno de la realidad virtual” (Nielsen & Laura, 2022).

Este invierno alcanza hasta finales de los 2000, cuando tres grandes empresas de Internet como son Facebook, YouTube y Google se interesan por el registro, procesamiento y distribución de los vídeos 360°.

El pionero de esta nueva industria fue *Google Street View*, lanzado por Google en 2007. Con esta aplicación el usuario puede elegir cualquier lugar del mundo y sumergirse en su entorno en 360°, lo que se considera una preparación cultural para todo el desarrollo posterior de los entornos inmersivos. A partir de aquí, todo crece y se desarrolla con la rapidez característica de la tecnología en las últimas dos décadas. Así pues, en 2015 se empiezan a obtener los primeros videos 360° experimentales gracias a la empresa Hero360, luego comprada por Google. Esta empresa es la pionera en desarrollar, a través de impresoras 3D, soportes para seis o diez cámaras, por lo general GoPro, para posteriormente obtener la imagen mediante la unión de las distintas tiras de vídeo de todas las cámaras (Nielsen & Laura, 2022). Pero tan solo dos años después, aparecen las primeras cámaras de video omnidireccional, las cuales ya incluían las distintas lentes que iban a captar el entorno. También en 2015 tanto Facebook como YouTube permiten la visualización y el *streaming* de contenido en 360°. Finalmente, la realidad virtual y el video 360° ya son una realidad.

2.2 Vídeos 360° o realidad virtual. Diferencias y similitudes

La mayor diferencia entre un vídeo 360° y la realidad virtual es que en el primer caso los objetos y elementos que se observan son reales, pertenecen a un ambiente real, mientras que en el segundo caso pertenecen, como su nombre indica, a un ambiente virtual, creado de forma digital, lo cual permite la interacción con ellos. Oliver Grau comenta que “las interfaces de RV generan la impresión de sumergirse en el espacio de la imagen, moverse e interactuar allí en ‘tiempo real’ e intervenir creativamente” (Grau, 2002).

Esta diferencia da pie a otras situaciones en las que el vídeo 360° y la realidad virtual se comportan de forma distinta (ver tabla 1). Por ejemplo, en la realidad virtual el usuario tiene cierta libertad de movimiento y de poder explorar dentro del mundo virtual en el que está sumergido, dificultando así también la creación de una historia. Mientras que en los vídeos 360° el autor de los vídeos tiene el control absoluto de aquello que se graba y, por tanto, se ve, y aquello que, por lo contrario, no se ve, siendo así mucho más sencillo formar una historia y dotar al producto final de coherencia narrativa (Hernández, 2022).

Aspectos	Realidad virtual	Video 360°
Fotografía	Entorno digital	Puede ser acción real o entorno digital
Movilidad	Mundo inmersivo a través del cual se puede mover	Limitado a lo captado a través de la cámara
Línea de tiempo	La duración puede variar en función de la exploración de cada usuario	Limitada a la duración del material grabado
Plataformas	Tan solo se puede vivir la experiencia a través de unas gafas de realidad virtual	Existen muchos reproductores compatibles con videos 360° (Incluido YouTube)
Historia	El usuario experimenta la historia y es parte de ella	El usuario sigue la historia ya definida

Tabla 1 Resumen diferencias entre realidad virtual y video 360°. Elaboración propia

Finalmente, cabe destacar que, tal y como se ha indicado, existe una diferencia de libertad de movimiento entre la realidad virtual y los vídeos 360°. Esto se conoce como grados de libertad (DoF). Así pues, en los vídeos 360° se tienen tan solo tres grados de libertad (3 DoF) frente a los seis (6 DoF) que se tienen con la realidad virtual, ya que se añaden tres grados al poder desplazarse dentro del entorno virtual (ver ilustración 4).

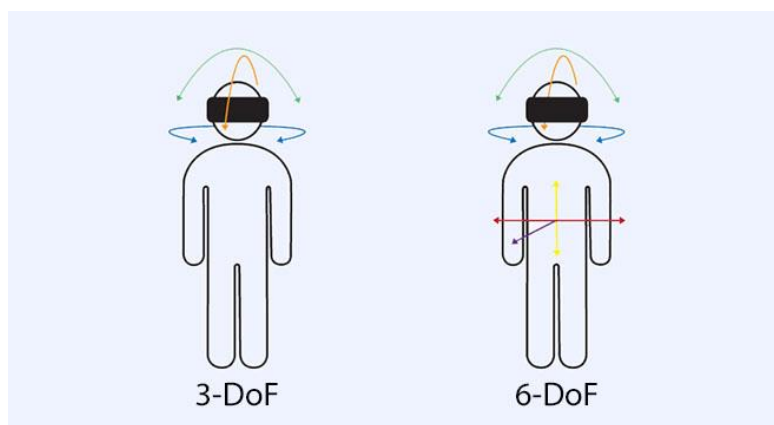


Ilustración 4 Diferencia 3 grados de libertad frente 6 grados de libertad. Fuente: <https://virtuallspeech.com/blog/degrees-of-freedom-vr>

2.3 La narrativa de los videos 360°

Uno de los aspectos más importantes de un vídeo es la narrativa de este. En los vídeos tradicionales esta narrativa depende del encuadre, de la escena y del montaje. Una de las formas más comunes de crear una narrativa es guiando la mirada del espectador a través del encuadre de la imagen. Sin embargo, en los vídeos 360° no se puede crear una narrativa con el encuadre de la imagen, pues el encuadre alcanza toda la escena.

Esta falta de encuadre es la que ha frenado a muchos directores de cine a dar el salto a los vídeos inmersivos. Jessica Brillhart, autora inmersiva para Google 360°, se pronuncia al respecto a través de su blog, diciendo que:

Para los realizadores el encuadre es algo difícil de dejar de lado y muchas veces nos hacemos preguntas equivocadas; ‘¿cómo podemos forzar al usuario a mirar hacia donde nosotros queremos y no hacia otro lugar?, ¿cómo podemos capturar su atención?’ Si estas son las preguntas más importantes que nos estamos formulando, significa que la RV no es el medio adecuado para nosotros. Terminaremos gastando una cantidad considerable de tiempo y energía tratando de controlar algo que fundamentalmente no podemos controlar (Brillhart, 7 de diciembre de 2015).

Y es que, dejando de lado el uso de un encuadre u otro para crear una narrativa, sí es posible crear una narrativa a través de los vídeos 360°, pero es una narrativa distinta a la narrativa del vídeo tradicional.

Una denominación que proponemos es “narrativas de encuadres” y “narrativas de entornos”. El cine, la televisión y el video tradicional construyen su narrativa a partir del marco o encuadre rectangular apaisado, por lo cual se pueden definir como “narrativa de encuadres o plana”; el video 360° y la RV, en cambio, se basan en la vista total de la imagen circundante, por lo que serán denominados como “narrativa de entornos o inmersiva”. (del Árbol y Nielsen, 2022, p. 27)

Además, cabe destacar que el video 360° no deja de estar en desarrollo y evolución, por lo que la inclusión de elementos que doten al contenido de un aspecto narrativo no deja de estar en una primera fase de desarrollo. Sin embargo, es muy probable que en un futuro el video 360° tenga muchos más elementos narrativos y no sea solo una narrativa de entornos.

Los entornos producen sensaciones, estamos tratando de encontrar la relación entre la construcción del espacio y el punto de vista para comunicar, porque es tan importante la acción dramática como el ambiente para lograr generar sensaciones en el usuario. (Nielsen & Laura, 2022)

2.3.1 El emplazamiento de cámara

En el cine convencional existe una gran variedad de tipos de plano y cada uno tiene una finalidad distinta. Por ejemplo, un plano general suele iniciar una secuencia y sirve para ubicar al espectador en el espacio. Por otro lado, un primer plano se utiliza para mostrar las emociones del personaje. Sin embargo, en los videos inmersivos, en general, y en los videos 360°, en particular, no existen planos ya que no existe un encuadre, pues la imagen abarca todo el entorno. Ante esta ausencia de planos, la posición de la cámara está condicionada, más que nunca, por la narratividad y claridad del producto final.

La escritura de una historia inmersiva se concibe desde el principio en función de su puesta, para crear una comunicación efectiva con el receptor que es, en última instancia, quien decide qué mirar. Es decir, desde el guion se piensa dónde se ubicará la cámara y cómo se distribuirá la acción (Nielsen & Laura, 2022).

En esta misma línea, se compara el emplazamiento de cámara del video tradicional frente al video 360°.

El emplazamiento de cámara en 360° implica la responsabilidad de situar al usuario en donde está la cámara, en el cine esto no pasa. Si decido poner la cámara como hice en La Casa tomada detrás de una cerradura de la puerta cancel estoy generando la sensación de que el usuario va a situarse en el lugar del mal. En cine uno pensaría que es un regocijo del director que quiso mostrar un plano raro, punto, nadie se pregunta por qué vemos desde ahí (Nielsen & Laura, 2022).

2.3.2 La iluminación

La desaparición del encuadre no solo tiene consecuencias a nivel narrativo, sino también a nivel técnico. Dado que grabar el entorno completo implica que no se tiene un “detrás de la cámara” donde se oculta tanto el equipo del rodaje como el material técnico como micrófonos y focos.

La iluminación es un aspecto fundamental para la inmersión en la acción, la expresividad de esta y también la continuidad entre planos y escenas. A la hora de elegir el tipo de

luz en la escena, se tienen distintas opciones, puede ser difusa (para reducir los contrastes) o dura (para destacar las figuras generando claroscuros), fría o cálida, con alto o bajo contraste. Dependiendo de lo que se quiera generar se utiliza un tipo de luz u otra.

En los vídeos 360° la puesta de luces es distinta respecto al vídeo tradicional, ya que como se ha comentado anteriormente, cualquier foco que se ponga en la escena va a ser grabado, lo que implica un proceso de posproducción en el que se elimine dicho foco. Ante esto, en los videos 360° existen los siguientes tipos de iluminación (Nielsen & Laura, 2022):

- Puesta de luces prácticas: Consiste en utilizar las luces que forman parte del entorno, evitando así la artificialidad que implica agregar un foco a la escena. Las fuentes lumínicas están dentro del entorno, es decir, forman parte de la diégesis (aparecer en la imagen y no desde fuera de esta). Con este tipo de iluminación se pretende tener una iluminación más realista. Néstor Almendros, destacado director de fotografía español, en su glosario de fotografía cinematográfica define: "*The available lights*: término del cine americano que define a la luz existente en un lugar no establecida por el iluminador" (1982, p.377). La puesta de luces prácticas o *available lights* era la única opción a utilizar en el vídeo inmersivo hasta que la tecnología avanzó lo suficiente.
- La técnica de *plates*: Gracias al avance de la técnica y el avance de las técnicas de retoque en posproducción, llegó la opción de utilizar la técnica de *plates*. Esta técnica consiste en utilizar una parrilla de luces en la parte superior del entorno para, posteriormente, poder borrar ese hemisferio de la imagen en posproducción. Dicho hemisferio es remplazado por una foto o un vídeo sin dicha parrilla de luces. Obviamente, esta técnica permite utilizar una iluminación más compleja y artística.

Comparando ambas técnicas, la técnica de *plates* permite utilizar la iluminación tal y como se usaría en el video tradicional, creando climas, sombras y contraluces controladas según las intenciones del director y el director de fotografía, creando, al fin y al cabo, narrativas a través de la iluminación. Sin embargo, tiene la desventaja de reducir el entorno para los personajes, ya que en algunos puntos del entorno habrá algún trípode con un foco, obstaculizando los movimientos de los personajes. Además, con el movimiento de dichos personajes también se pueden generar algunas sombras

no deseadas ya que pueden parecer demasiado antinaturales. Finalmente, utilizar la técnica de *plates* extiende el tiempo de posproducción que se tiene que sustituir el hemisferio dónde se encuentren ubicadas las luces (Nielsen & Laura, 2022).

Por otro lado, que utilizar la puesta de luces prácticas es más sencillo, otorga naturalidad al relato y ahorra tiempos de posproducción. Esta naturalidad aportada por las *available lights* es la que ha decantado la decisión de optar por este tipo de iluminación en este TFG, tal y como se comentará posteriormente.

2.4 Los vídeos 360° en el sector turístico

El desarrollo de los videos 360° ha tenido una gran influencia en diversos sectores, principalmente en el sector turístico y en el sector del marketing, ya que durante años tanto los lugares de interés turístico, museos, parques, hoteles como los productos de una empresa eran mostrados a través de vídeos e imágenes. Hoy en día todo se puede mostrar a través de un vídeo 360°, lo que significa un acercamiento del producto con el usuario interesado (tour360, 2022).

Además, debido a la reciente pandemia de la Covid-19, sobre todo en el periodo de restricciones, el uso de vídeos 360° en el sector turístico también tuvo auge; ya que el turismo digital permite visitar una gran variedad de sitios sin salir de casa, simplemente contando con un ordenador y conexión a Internet. El 360° es la posibilidad de mostrarle a alguien algo, un lugar en el que nunca estuvo o en el que nunca va a poder estar (Leyrado, 2016).

A través del turismo digital lo que se busca es tener el máximo realismo posible, con el objetivo de que el usuario se sienta como si realmente hubiera viajado hasta ese lugar. La mejor forma de crear este realismo es utilizando vídeos inmersivos, ya sea con realidad virtual o con vídeos 360°.

Los videos 360° son una alternativa rápida y económica para aquellas personas que desean viajar y visitar lugares nuevos, pero que, por distintos motivos personales, como enfermedades, o motivos económicos no pueden desplazarse hasta ciertos lugares.

Finalmente, también cabe destacar que los vídeos 360° y los tours virtuales creados a partir de estos, son una alternativa sostenible. A veces, una de las mayores consecuencias del turismo es el sobre turismo, es decir, el hecho de que ciertos puntos

turísticos tengan de forma anual una exagerada cantidad de turistas. Algunos lugares históricos como ruinas, murallas o castillos se pueden deteriorar con el paso del tiempo y debido a la cantidad excesiva de personas que los visitan, por lo que a través de los vídeos 360° estos lugares podrían quedar mejor conservados al sufrir una deterioración menor (tour360, 2022).

3. Codificación en *tiles*

3.1. ¿Qué es?

La codificación en *tiles* es una técnica innovadora que divide un espacio continuo en secciones más pequeñas para mejorar la generalización en comparación con un enfoque basado en una única cuadrícula. El objetivo principal de esta técnica es la creación de varias cuadrículas superpuestas o *tilings*. Cada *tiling* está formado por diversos *tiles*. Con esto se consigue que, posteriormente, para cada valor de una muestra tan solo sea necesario determinar en qué cuadrículas se encuentra, para así poder representar el valor continuo original mediante un vector de índices enteros que almacena cada una de las cuadrículas en las que se encuentra la muestra (Cook, 2017).

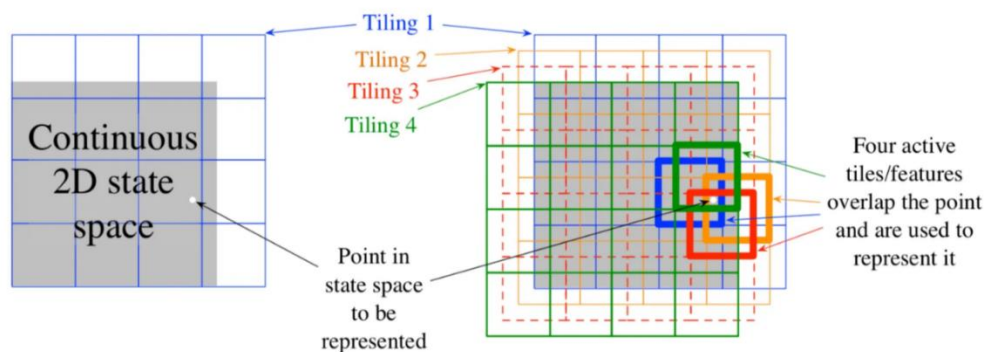


Ilustración 5 Codificación por tiles. Fuente: <https://towardsdatascience.com/reinforcement-learning-tile-coding-implementation-7974b600762b>

Por ejemplo, en la ilustración 5 se tiene un ejemplo de cómo se aplicaría la codificación por *tiles* en un espacio continuo en 2D. En la imagen de la izquierda se tiene representado tanto el espacio de estado, representado por una cuadrícula simple en 2D con líneas azules, como el espacio de características, representado por un área gris (cabe destacar que no es estrictamente necesario que el espacio de características cubra todo el espacio de estado). Como se observa en la imagen, hay un punto blanco que es la muestra de este ejemplo y representa el valor de estado, tomando valor 1 para la cuadrícula a la que pertenece y valor 0 para el resto de las cuadrículas.

En cuanto a la imagen de la derecha, se observa que se tienen distintos *tilings* que forman la representación, cada uno con diferentes *offset*. Debido a esto, en esta ocasión el punto blanco se encuentra en cuatro *tiles* distintos (Zhang, 2019).

Por lo tanto, en la Ilustración 5, se observa, a la izquierda de la imagen, la codificación en *tiles* con un solo *tiling* donde la muestra se representa únicamente por el *tile* en el que cae, perteneciendo así a un solo *tile*. Sin embargo, al utilizar múltiples *tilings*, cada muestra se representa con diversos *tiles*, los cuales pertenecen a cada uno de los distintos *tilings*. Esta técnica tiene más potencial porque cada valor de estado puede compartir algunos *tiles* y también pertenecer a diferentes *tiles* al mismo tiempo, lo cual es esencial para la generalización.

3.2 La técnica *tile coding* en los videos 360°

El inicio de la comercialización de los cascos de realidad virtual ha generado un aumento de interés en la inclusión en la industria del video inmersivo y el video 360°. Del mismo modo, también ha aumentado el interés por parte de los grupos de investigación, los cuales han puesto el foco en trasladar las técnicas utilizadas en el video convencional al video inmersivo, lo cual no es un proceso sencillo, ya que, aunque los principios básicos son los mismos, el video inmersivo cuenta con una serie de desafíos específicos que deben afrontarse.

Gran parte de dichos desafíos pertenecen al apartado de la transmisión en vivo o *streaming*. Esto se debe a que el hecho de utilizar una vista omnidireccional en lugar de una vista en 2D requiere un aumento notorio del ancho de banda para poder transmitir con la misma calidad con la que se retransmitiría un video convencional. Es por eso por lo que la codificación en *tiles* se ha convertido en un estándar de los videos 360°, ya que utilizando esta técnica se reduce el flujo de datos de las transmisiones. Se divide la esfera en diversos *tiles*, en función de un esquema de proyección previamente definido, y cada *tile* puede ser descargado como un objeto separado del resto de *tiles*. De esta forma, se puede realizar una concentración de recursos en aquellos *tiles* que se encuentran dentro de su campo de visión, y reduciendo los recursos de los *tiles* que están fuera del campo de visión, los cuales tendrán una resolución baja o ni siquiera serán descargados (ver ilustración 6). Esto tiene como consecuencia directa que se pueda garantizar una mínima calidad de experiencia (QoE) del usuario utilizando la menor cantidad de recursos posible (Yaqoob, Bi, & Muntean, 2020).

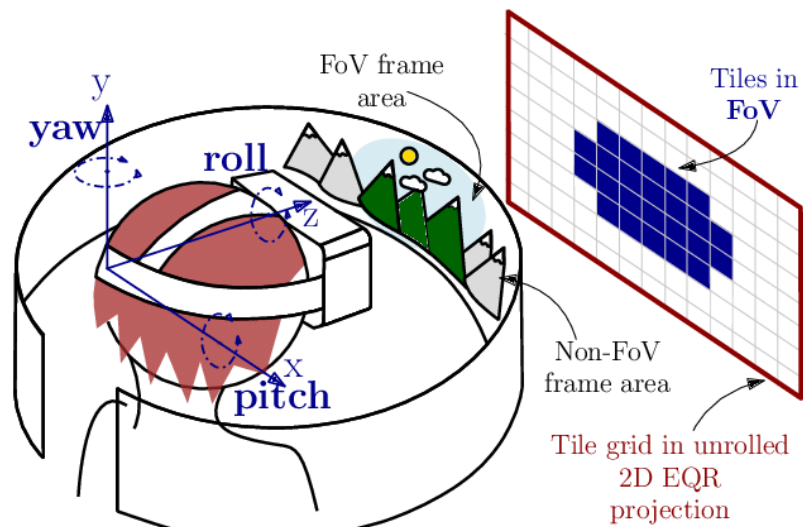


Ilustración 6 Ejemplo del FoV en la codificación por tiles. Fuente: https://www.researchgate.net/figure/Tiled-FoV-mapping-of-a-users-3DoF-pose-in-the-EQR-projection-of-a-360-video-frame_fig1_329102051

Esta técnica tiene dos aspectos claves para su correcto funcionamiento. En primer lugar, se requiere una predicción precisa de qué región de *tiles* será la próxima en ser visualizada por el usuario, prediciendo así en qué lugar se fijará el usuario, lo cual es un tema de investigación bastante complejo y que sigue en constante desarrollo. En segundo lugar, el diseño del esquema de *tiles* también es un factor importante, tanto en la eficiencia de compresión del esquema de codificación de video como en la calidad de experiencia del usuario (Yaqooh, Bi, & Muntean, 2020).

3.2.1 El problema de la distorsión

Debido a que los filtros y las herramientas de codificación mayoritariamente están basados en imágenes en 2D, el contenido almacenado en una esfera debe ser proyectado en un plano para ser procesado y codificado. Aparece así el problema de distorsión geométrica al que se llevan enfrentando durante años los cartógrafos al representar un mapa de la Tierra. El proceso de proyectar una imagen esférica en un plano conlleva inevitablemente algún tipo de distorsión. No obstante, la diferencia entre los videos 360° y la cartografía es que en los videos 360° la proyección no es la única fase del proceso que añade distorsión a la imagen.

En primer lugar, se tiene la distorsión añadida en la adquisición de la imagen, ya que el hecho de obtener las imágenes a partir de múltiples cámaras, para luego realizar el

*stitching*¹, puede introducir diversos problemas con los bordes de las imágenes grabadas por cada una de las cámaras. Algunos de estos problemas abarcan desde la falta de información (por ejemplo, objetos que desaparecen) hasta cambios en la exposición de la imagen capturada por cada cámara (notando unas zonas con mayor o menor luz que otras, ya que han sido capturadas con cámaras distintas) o desalineamiento de los bordes de las imágenes de cada una de las cámaras. Dichos problemas son más frecuentes en los polos de la imagen, dado que la mayoría de las cámaras 360° no tienen la capacidad de capturar dichos polos y estos se reconstruyen en la posproducción del vídeo. Aunque sea menos frecuente, también pueden aparecer problemas de discontinuidades temporales, con objetos que aparecen y desaparecen, o deformaciones de ciertos objetos, si estos están ubicados en las zonas correspondientes a las áreas de unión entre las distintas cámaras.

Una vez que se ha capturado la imagen omnidireccional, es necesario convertirla en una representación plana para su codificación y almacenamiento. Posteriormente, se divide en *tiles* para facilitar la transmisión basada en *tiles*. La combinación de la proyección del mapa y los bordes de los *tiles* generan patrones de distorsión que interactúan entre sí. Como resultado, la forma y gravedad de los efectos de distorsión geométrica dependen en gran parte de la proyección y del esquema de *tiles* utilizados, lo cual es crucial para lograr una compresión eficiente del video omnidireccional. Una buena elección del sistema de proyección asegurará una compresión eficiente.

3.2.1.1 Proyecciones

- *Equirectangular Projection* (ERP): Es la proyección más común. Tal y como su nombre indica divide la esfera en una serie de rectángulos. El problema de esta proyección reside en que se codifica con más píxeles los polos que el ecuador, lo cual es un desperdicio dado que el usuario suele prestar más atención al ecuador que a los polos.

En la ilustración 7 se muestra un ejemplo de la proyección equirrectangular, donde las distintas letras hacen referencia a las distintas zonas de la esfera. Así pues, al aplanar la esfera se tiene la parte frontal (F), las laterales izquierda y derecha (L y R), la parte trasera (B) y los polos superior e inferior (U y D).

¹ Proceso mediante el cual se combinan las múltiples imágenes captadas por las distintas cámaras para formar una única imagen

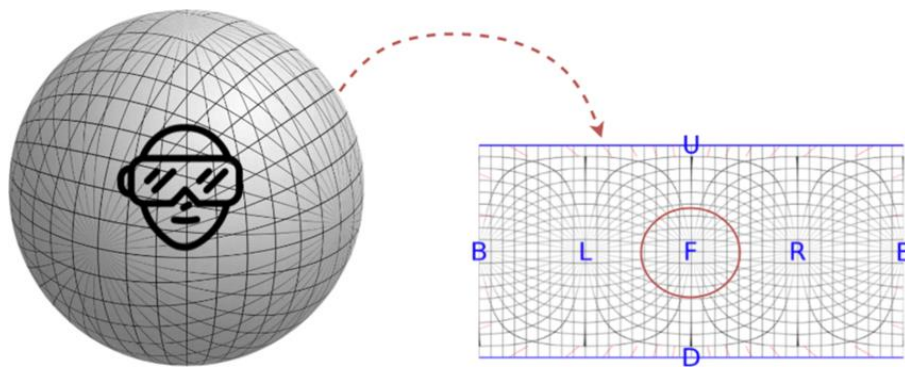


Ilustración 7 Proyección equirectangular. Fuente: https://www.researchgate.net/figure/Equirectangular-projection-On-the-left-is-shown-is-the-spherical-representation-of-the_fig2_340243249

- *Dyadic projection*: Busca resolver el problema del sobre muestreo en los polos a partir de la reducción de dicho muestreo para ángulos superiores a $\pi/3$ desde el ecuador.
- *Barrel projection*: Codifica los rectángulos superior e inferior del ERP como círculos, reduciendo así el número de píxeles de estas zonas
- *Polar square projection*: Sistema parecido a la proyección anterior, pero cambiando la codificación de los polos por dos cuadrados.
- *Equal-area cylindrical projection*: Consiste en reducir la altura de los tiles en función de la latitud (ver ilustración 8).

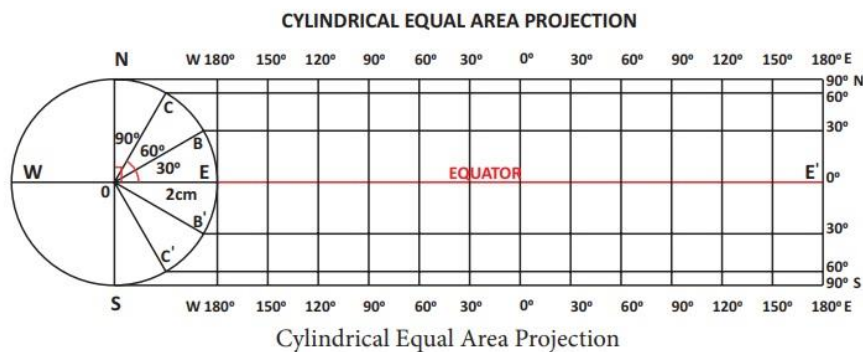


Ilustración 8 Diagrama de la equal-area cylindrical projection. Fuente: https://www.brainkart.com/article/Cylindrical-Equal-Area-Projection---Lambert---s-Cylindrical-Equal-area-projection_41153/

- *Nested Polygonal Chain Mapping (NPCM)*: Parte de la proyección ERP y se aproxima a la densidad de muestreo óptima. Es otra técnica basada en la reducción del muestreo
- *Cubic Mapping Projection (CMP)*: Esta proyección se está extendiendo hoy en día. Consiste en crear un cubo alrededor de la esfera y proyectar un rayo desde el centro hacia fuera (ver ilustración 9). Por tanto, cada rayo se cruza en un único

punto con la superficie de ambas figuras. Estos puntos definen el mapeo de proyección. Esta técnica es la utilizada por Facebook ya que presenta más eficiencia, en términos de compresión, que la ERP.

También hay que destacar que el uso de polígonos con un número elevado de caras como pueden ser los octaedros, dodecaedros rómbicos o icosaedros, reduce los problemas de distorsión de los lados. Esto se debe a que las aristas de estos polígonos requieren ser menos estiradas que las aristas de un cubo. Sin embargo, los polígonos de muchas caras pese a tener menos distorsión en la proyección, tienen un mayor número de límites discontinuos.



Ilustración 9 Diagrama de la Cubic Mapping . Fuente: Projection.<https://www.trekview.org/blog/2021/projection-type-360-photography/>

- *Rotated Sphere Projection (RSP)*: Consiste en desplegar la esfera bajo dos ángulos de rotación, y unir las partes como si fuera una pelota de béisbol. Esta técnica parte de la ERP y aumenta su eficiencia de codificación.
- *ClusTile*: Esta técnica está basada en el comportamiento del espectador. Así pues, crea un diseño de un conjunto de *tilas* basado en el comportamiento del espectador.

Pese a que se han comentado algunas de las proyecciones, se resumen se resumen las características más importantes de las distintas proyecciones en la tabla 2 (Chiariotti, 2021).

Proyección	Geometría	Principales ventajas y problemas
<i>Equirectangular</i>	Cada rectángulo tiene el mismo ángulo sólido ²	Sobre muestreo en los polos
<i>Dyadic</i>	Equirectangular con reducción del muestreo en los polos	Distorsión en los polos
<i>Barrel</i>	La esfera esta mapeada en un cilindro	Distorsión en los bordes de las imágenes captadas por las distintas cámaras

² El ángulo sólido es el ángulo espacial que abarca un objeto visto desde un punto dado, que se corresponde con la zona del espacio limitada por las rectas proyectantes desde el objeto hacia el observador (Wikipedia, s.f.)

<i>Polar square</i>	En forma de barril, mapeando los polos como cuadrados	Distorsión en los polos
<i>Equal-area cylindrical</i>	Equirrectangular, pero la altura del <i>tile</i> depende de la latitud	Reduce el sobre muestreo de los polos
<i>Latitude adaptive</i>	Equirrectangular, pero la latitud depende del número de <i>tiles</i>	Reduce el sobre muestreo de los polos
<i>Rhombic mapping</i>	Similar a la anterior pero los <i>tiles</i> son rombos	Eficiente división de <i>tiles</i>
<i>Nested polygonal chain</i>	Equirrectangular, pero con una reducción de muestreo	Reduce el sobre muestreo de los polos
<i>Cubic mapping</i>	Proyección de una esfera a un cubo	Mayor eficiencia, reduce la distorsión en los polos, pero tiene distorsión en los bordes de las imágenes captadas por las distintas cámaras
<i>Equirectangular cubic mapping</i>	Equirrectangular, pero en las caras de un cubo	Reduce la distorsión en los bordes de las caras
<i>Other solids</i>	Proyecciones en otros polígonos con más caras	Menor distorsión de proyección, pero mayor distorsión en los bordes
<i>Variable tile shape</i>	El tamaño de los <i>tiles</i> se adapta al contenido	Menor distorsión, pero mayor complejidad en la codificación y decodificación
Rotated sphere	Se forma un despliegue similar al de una pelota de béisbol	Aumenta la eficiencia en la codificación y tiene poca distorsión en los bordes de las imágenes captadas por las distintas cámaras
ClusTile	Muestreo adaptativo basado en el comportamiento del espectador	Poca distorsión, pero la codificación y decodificación es compleja

Tabla 2 Resumen de las características de las distintas proyecciones. Elaboración propia con fuente: <https://arxiv.org/abs/2102.08192>

4. Streaming Adaptativo basado en HTTP (HAS)

4.1 Tecnologías disponibles

Existen tres tipos de tecnologías distintas a la hora de hacer *streaming*:

-*Streaming* con descarga completa: hace referencia al *streaming* tradicional y consiste en realizar una descarga completa del fichero a reproducir. Hasta que dicho fichero no es descargado completamente, no se inicia la reproducción.

-*Streaming* con descarga progresiva: hace uso de un *buffer* de datos el cual va almacenando el material descargado. Esto permite que se pueda iniciar la reproducción sin que se haya realizada la descarga completa, tan solo teniendo una parte inicial del fichero a reproducir. La idea es que se pueda continuar la reproducción mientras haya contenido en el *buffer*. Pueden causar saltos y pausas en la reproducción debido a algún fallo en la descarga, o si en algún momento no hay contenido en el *buffer*.

-*Streaming* adaptativo: su funcionamiento es similar, al anterior, pero en este caso, se utilizan técnicas para asegurarse de que el *buffer* no se quede sin contenido que reproducir. Se pueden utilizar técnicas en ambos extremos. Por un lado, la fuente puede adaptar la tasa de transmisión según el *feedback* recibido del estado de la reproducción del receptor. Por otro lado, el receptor puede utilizar, por ejemplo, técnicas HAS y, descargarse en cada momento fragmentos del contenido (llamados *chunks*) de diferente calidad, según ciertas condiciones y parámetros de red y del propio dispositivo de recepción y reproducción.

4.1.1 *Streaming* con descarga progresiva tradicional

En este tipo de tecnología se utiliza los protocolos como *User Datagram Protocol* (UDP) y *Real Time Transport Protocol* (RTP). Utilizando dichos protocolos, los clientes se conectan al servidor y mantienen la sesión hasta que se desconectan de este servidor. Durante estas sesiones los clientes pueden reproducir o pausar el contenido, entre otras acciones.

Estos protocolos tienen ciertas desventajas, entre las que se destacan las siguientes (Sangameshwarkar, Study of video streaming standards, 2017):

-Esta tecnología requiere de un servidor especializado el cual hay que configurar y mantener, por lo que en caso de usar una implementación a gran escala resultaría costoso.

-El propio servidor tiene que realizar un seguimiento del estado de cada sesión de transmisión, lo que aumentará el coste del servidor.

-Si la red es inestable, se puede producir pérdida de paquetes o retrasos en la transmisión de estos.

-Se utilizan protocolos de transporte como RTP o UDP, que pueden sufrir problemas de bloqueo en los cortafuegos de las empresas, así como tener problemas debidos al uso de NAT.

4.1.2 *Streaming* con descarga progresiva basada en HTTP

Muchos de los sitios web más populares como YouTube, Vimeo o Myspace, utilizan o han utilizado la tecnología de descarga progresiva basada en HTTP, cuyas principales características son las siguientes (Sangameshwarkar, Study of video streaming standards, 2017):

- Está especializado para la entrega de paquetes
- Utiliza como protocolo de transporte de datos el protocolo TCP, lo cual simplifica los problemas del cortafuegos y NAT.
- Como desventaja, esta tecnología no permite cambiar la calidad del video en función de la congestión de la red. Así pues, todos los clientes reciben el mismo video con la misma codificación, pese a todos los cambios en el ancho de banda que el cliente pueda sufrir a lo largo del tiempo.

4.1.3 *Streaming* adaptativo basado en HTTP (HAS)

Es la tecnología más usada a día de hoy debido a las grandes ventajas que tiene como el uso de técnicas para asegurar que el *buffer* de reproducción no se quede sin contenido, como ya se ha comentado anteriormente. Esta tecnología es compatible con el uso de cortafuegos y NAT. Otras de sus ventajas son las siguientes (Sangameshwarkar, Study of video streaming standards, 2017):

- Entrega escalable al permitir utilizar los servidores HTTP extendidos por todo el mundo.
- Favorece la convergencia fijo-móvil.
- Favorece la convergencia de servicios.
- Mejora la calidad de la experiencia (QoE) de los usuarios.

Por otro lado, la tecnología HAS es más compleja que la transmisión, lo cual implica un coste adicional en el almacenamiento, codificación y mantenimiento. En la tabla 3 se recoge una comparativa de las principales características de las tecnologías de transmisión comentadas.

Características	Streaming con descarga progresiva tradicional	Streaming con descarga progresiva basada en HTTP	Streaming adaptativo
Protocolo de control de la transmisión	RTSP o RTMP	HTTP	HTTP
Protocolo de transmisión	RTP o UDP	TCP	TCP
Control sobre la transmisión	Si	No	Si
Problemas con el uso de cortafuegos y NAT	Si	No	No
Capacidad de adaptación al ancho de banda	Sí	No	Sí
Beneficios claves	Se eliminan los problemas de falta de contenido en el <i>buffer</i> de recepción	Especializado para la entrega de paquetes	Entrega escalable al permitir utilizar los servidores HTTP extendidos por todo el mundo
Desventajas	Se necesita una configuración de la red para la habilitación de puertos y protocolos (RTP, UDP), que a menudo son bloqueados por los cortafuegos corporativos	No permite cambiar la calidad del video en función de diversos aspectos, como la congestión de la red	Requiere un desarrollo adicional del lado del cliente y, también, del lado del servidor

Tabla 3 Resumen de las características de las tecnologías de transmisión. Elaboración propia con fuente: <https://42jaiio.sadio.org.ar/proceedings/simposios/Trabajos/EST/19.pdf>

4.2 Sistemas basados en tecnologías HAS (*Streaming* Adaptativo basado en HTTP)

Las tecnologías HAS permiten cambiar los flujos de distintas calidades de manera adaptativa para optimizar la experiencia del usuario. Como tecnología basada en HTTP, no se requiere un servidor de transmisión, por lo que toda la lógica de conmutación entre una calidad u otra reside en el reproductor. Para realizar la distribución al usuario, primero se convierte el contenido original a varios flujos en diferentes calidades (a diferentes velocidades de datos) y se divide cada flujo en los mismos fragmentos cortos, generalmente entre 5 y 10 segundos de duración. El usuario recibe información sobre los distintos flujos disponibles a través de un archivo llamado *manifest*. Cuando comienza a reproducir contenido, el usuario solicita las partes del contenido que tienen

la calidad más baja. Si el usuario nota que la velocidad de descarga es más rápida que el *bitrate* del segmento que está viendo, entonces pedirá los próximos fragmentos con una calidad superior.

Actualmente existen cuatro sistemas basados en tecnología HAS.

1. HLS (HTTP Live *Streaming*) de Apple. (Apple, 2017)

Fue implementado en 2009 su última versión es la 7 y está especificada en el RFC-8216. Es uno de los protocolos líderes para la distribución de contenido multimedia mediante HAS. Uno de sus puntos fuertes es su gran compatibilidad con reproductores, dispositivos móviles tanto de iOS como de Android, navegadores web y servidores de transmisión. Además, esta tecnología permite la encriptación con tal de proteger el contenido, siendo una de sus desventajas la latencia elevada cuando se utiliza para una transmisión en vivo (Telefónica, s.f.).

Los fragmentos cortos generados de distintas calidades se cargan en un servidor HTTP junto con un archivo llamado *manifest* con una extensión .M3U8 que dirige al reproductor a archivos adicionales para cada uno de los flujos codificados. (Rai, 2020)

Se envía al cliente el archivo *manifest* con la lista de flujos disponibles, codificados a diferentes velocidades de bits, utilizando una lista de reproducción M3U extendida.

En la tabla 4 mostrada a continuación se indica un resumen de esta tecnología.

<i>Manifest</i>	Archivo con extensión .M3U8
Codificación de video	H.264
Codificación de audio	MP3 o HE-ACC
Tipo de contenedor	MPEG-2
Tipo de servidor	No necesita un servidor especial. Sólo un servidor HTTP convencional

Tabla 4 Resumen HLS. Elaboración propia con fuente:
<http://www.ijlera.com/papers/v2i6/12.201706344.pdf>

2. MSS (MS Smooth *Streaming*) de Microsoft (Microsoft, 19)

En 2008, Microsoft anunció el lanzamiento del nuevo *Internet Information Server* (IIS) 7.0 con la nueva inclusión de una técnica de *Streaming* adaptativo basado en HTTP, llamada *Smooth Streaming*, con el objetivo de proporcionar una entrega fluida de contenido.

Su cliente basado en *Silverlight*³ detecta de forma constante qué representación del contenido se adapta mejor a las condiciones de cada instante, teniendo en cuenta el ancho de banda disponible, el uso de la CPU y la resolución de la ventana de reproducción. Esta técnica de transmisión fue la utilizada en los Juegos Olímpicos de Beijing de 2008, el Campeonato de Wimbledon en 2009 o los Juegos Olímpicos de Invierno de Vancouver en 2010. Allí se demostró que el *streaming* adaptativo basado en HTTP funcionaba perfectamente en transmisiones en vivo de grandes eventos, que implican las visualizaciones de millones de espectadores. Cabe destacar que el hecho de que Microsoft incluya el uso de la CPU como factor determinante para la decisión de cambio de flujo, es un aspecto muy destacable cuando se utiliza un móvil o una *tablet*, ya que estos dispositivos tienen capacidades limitadas tanto de CPU como de GPU. Esto significa que, si la utilización de la CPU es alta, el cliente reduce la calidad y la resolución del flujo, reduciendo así aún más las necesidades de rendimiento de la CPU del proceso de decodificación y, por tanto, se garantiza una decodificación continua sin pérdidas. *Microsoft Smooth Streaming* aprovecha tres tipos de archivos diferentes para su sistema de *streaming* (Mueller, Bitmovin, 2015) :

- Archivos MP4 fragmentados para contenido multimedia, con extensiones .ismv, para video; y .isma, para audio.
- Archivo *manifest* del servidor con extensión .ism. Describe las relaciones entre las pistas multimedia, las tasas de bits y los archivos en el disco
- Archivo *manifest* del cliente con extensión .ismc. Describe al cliente las transmisiones disponibles, los códecs utilizados, las tasas de bits codificadas, las resoluciones de video, los marcadores, los subtítulos, etc.

Los archivos MP4 fragmentados se basan en el formato *ISO Base Media File Format* (IBMF), el cual se organiza básicamente utilizando bloques como unidades para datos y metadatos. En el caso de las transmisiones, MP4 ofrece la posibilidad de dividir los metadatos y los datos multimedia de una transmisión continua en varios fragmentos (ver ilustración 10), cada uno de los cuales consiste en un bloque de metadatos y un bloque de datos multimedia, también etiquetado como MP4 fragmentado (fMP4) en el contexto de *Smooth Streaming*. Por lo tanto, es posible almacenar segmentos multimedia separados, que

³ Silverlight es una herramienta de desarrollo para crear experiencias de usuario atractivas e interactivas para aplicaciones web y móviles (Microsoft, s.f.)

corresponden a uno o más grupos de imágenes (GOP), mediante un bloque de fragmentos de película (moof) y un bloque de datos multimedia (mdat).

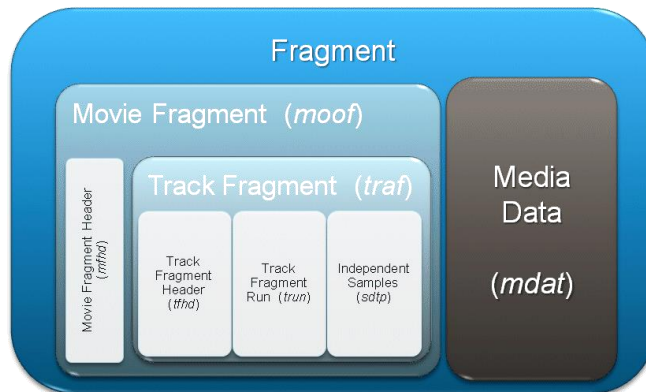


Ilustración 10 Estructura fragmento Smooth Streaming. Fuente: <https://alexzambelli.com/blog/2009/02/10/smooth-streaming-architecture/>

Todos los fragmentos de película de la misma representación (es decir, del mismo archivo de video con una calidad determinada) se almacenan juntos en un archivo MP4 que permite el acceso aleatorio (ver ilustración 11). Este archivo comienza con un bloque de metadatos de película (moov) que contiene información de metadatos para todo el archivo, seguido de los diferentes fragmentos. Al final del archivo, hay un bloque de fragmentos de movimiento (mfra) que permite un acceso aleatorio rápido a los diferentes fragmentos (Mueller, Bitmovin, 2015).

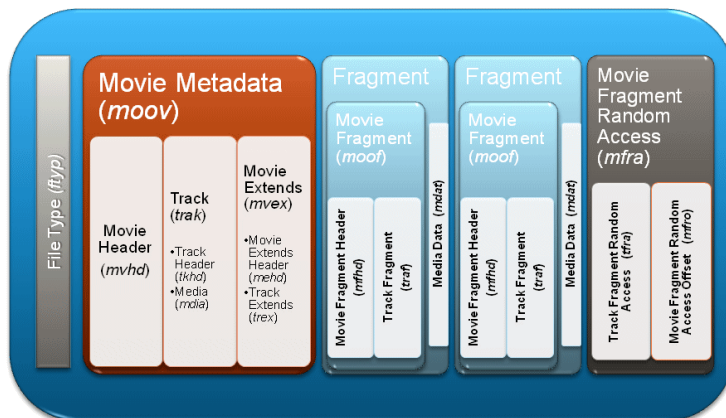


Ilustración 11 Estructura Smooth Streaming. Fuente <https://alexzambelli.com/blog/2009/02/10/smooth-streaming-architecture/>

Los clientes, solicitan los fragmentos al servidor web ISS, que interpreta la petición HTTP GET entrante y responde con el fragmento de la reproducción apropiado. En la tabla 5 mostrada a continuación se indica un resumen de esta tecnología (Mueller, Bitmovin, 2015).

<i>Manifest</i>	Archivo XML con extensión ism/ismc
Codificación de video	VC-I o H.264
Codificación de audio	ACC o WMA
Tipo de contenedor	MP4 con extensión ismv/isma
Tipo de servidor	Servidor HTTP

Tabla 5 Resumen MSS. Elaboración propia con fuente: <http://www.ijlera.com/papers/v2-i6/12.201706344.pdf>

3. HDS (HTTP Dynamic Streaming) de Adobe (Adobe, s.f.)

HDS es un protocolo de transmisión basado en Flash que representa la evolución de RTMP (*Real-time Messaging Protocol*) por parte de Adobe, aunque, a día de hoy, está en desuso. El proceso de creación y entrega de una transmisión HDS sigue las mismas líneas que la transmisión de las técnicas HAS anteriores. En primer lugar, los archivos MP4 deben pasarse a MP4 fragmentado (archivos F4F), los cuales contienen audio, vídeo y metadatos. Tal y como ocurre en las otras técnicas de transmisión HAS, la técnica HDS codifica distintas versiones con distintas calidades del mismo archivo de vídeo. Además, los vídeos se dividen en segmentos más cortos de unos pocos segundos de duración. Finalmente, los segmentos del vídeo se envían a los clientes conforme estos solicitan su transmisión a través de Internet mediante HTTP (haciendo uso de la información almacenada en el archivo *manifest*).

4. DASH (Dynamic Adaptive Streaming over HTTP) de MPEG (MPEG, 2014)

Siendo publicado en 2012 y actualizado en 2014, MPEG-DASH es el primer estándar de HAS internacional y abierto, desarrollado por MPEG (*Moving Picture Experts Group*). Este estándar es apoyado y promocionado por el DASH-IF (*Dash Industry Forum*), formado por empresas como Microsoft, Netflix, Google, Samsung, Adobe, etcétera.

El objetivo de la creación del estándar DASH era reemplazar las múltiples soluciones anteriormente comentadas, con tal de ofrecer la interoperabilidad y convergencia necesaria para asegurar el crecimiento a gran escala del *streaming* adaptativo basado en HTTP.

En dicho estándar se definen los siguientes aspectos:

- El formato del fichero índice
- El formato de los segmentos, que pueden contener cualquier tipo de datos. Sin embargo, la especificación proporciona guías y formatos específicos para ser

usados con dos tipos de contenedores: *ISO base media file format* o BMFF (MP4) o *MPEG-2 Transport Stream*.

Sin embargo, no se definen los siguientes aspectos:

- Ningún sistema, protocolo de transporte, códec, o aspectos por el estilo
- El mecanismo de envío del MPD, la codificación del contenido multimedia ni el comportamiento del cliente

En la tabla 6 mostrada a continuación se indica un resumen de esta tecnología.

<i>Manifest</i>	Fichero con extensión .mpd (Media Presentation Description)
Codificación de video	Cualquier códec
Codificación de audio	Cualquier códec
Tipo de contenedor	MP4 o MPEG-2
Tipo de servidor	Servidor HTTP

Tabla 6 Resumen DASH. Elaboración propia con fuente <http://www.ijlera.com/papers/v2i6/12.201706344.pdf>

4.2.1 Comparación de los cuatro sistemas HAS

En la tabla 7 se recoge una comparativa de estas cuatro técnicas de *Streaming* adaptativo basado en HTTP.

Características	HDS	HLS	MSS	DASH
Múltiples canales de audio	No	Sí	Sí	Sí
Subtítulos	Sí	Sí	Sí	Sí
Inserción efectiva de anuncios	No	Sí	No	Sí
Cambio rápido de canal	Sí	No	Sí	Sí
Cifrado de contenido	Sí	Sí	Sí	Sí
Multiplexación de audio y video	Sí	Sí	No	Sí
Segmentos MPEG-2 TS	No	Sí	No	Sí

Tabla 7 Resumen características HDS, HLS, MSS. Elaboración propia. Fuente: <https://bitmovin.com/mpeg-dash-vs-apple-hls-vs-microsoft-smooth-streaming-vs-adobe-hds/>

Comparando el sistema DASH con el resto de las tecnologías HAS se tienen las siguientes ventajas:

- Es un estándar gratuito, independiente y ratificado internacionalmente por MPEG/ISO.
- Soportado por Netflix, YouTube o Qualcomm, entre otros.

Además, las siguientes características las comparte con HLS, pero son una ventaja frente a MSS y HDS:

- El uso de DASH se está estandarizando con reproductores HTML5, incluyendo protección DRM.
- Admiten múltiples tipos de contenedores, así como los códecs más extendidos.
- Ofrece soporte para subtítulos y anuncios publicitarios
- Está siendo adoptada por otros estándares y organismos como HbbTV (*Hybrid Broadcast Broadband TV*).

4.3 MPEG-DASH

4.3.1 El fichero índice y sus componentes

En MPEG-DASH el fichero índice se conoce como *Media Presentation Description file* (MPD). Dicho archivo es un fichero XML que contiene los metadatos de aquellos contenidos que se han generado y almacenado en el servidor e indican al cliente como acceder a cada una de sus partes.

Al igual que ocurre en el resto de las tecnologías HAS, una vez el cliente se descarga el fichero índice, con esta información, se ejecuta un algoritmo de selección de segmentos para evitar interrupciones en el proceso de reproducción. Gracias a dicho algoritmo se maximizará la calidad promedia durante la reproducción y se minimizará el retardo inicial de reproducción.

La estructura de un fichero mpd sigue un modelo jerárquico formado por cinco niveles (ver ilustración 12), siendo sus principales componentes los siguientes:

- *Period*: Hacen referencia a distintas etapas temporales en las que se divide el contenido. Por ejemplo, una película podría estar dividida en tres *Periods* con el objetivo de insertar dos bloques de anuncios entre los *Periods*, constituyendo los propios anuncios también *Periods*, teniendo en total cinco *Periods*. Si el contenido no va a ser interrumpido podría tener un único *Period*.
- *Adaptation Set*: Hacen referencia a cada uno de los componentes del contenido dentro de cada *Period*. Por ejemplo, se puede tener un *Adaptation Set* referente al flujo de vídeo del contenido y otros dos referentes a flujos de audio, teniendo un *Adaptation Set* para el audio en castellano y otro para el audio en inglés.

- **Representation:** Hace referencia a cada una de las calidades con las que se tiene el contenido de cada *Adaptation Set*. Es decir, se tiene una *Representation* por cada una de las calidades almacenadas y ofrecidas por el servidor.
- **Segment:** Por último, los *Segment* hacen referencia a cada uno de los fragmentos en los que se ha dividido el contenido de cada *Representation*. Todos los *Segment* tienen la misma duración y son solicitados (o no) en función del criterio establecido, por ejemplo, en función del ancho de banda del usuario.

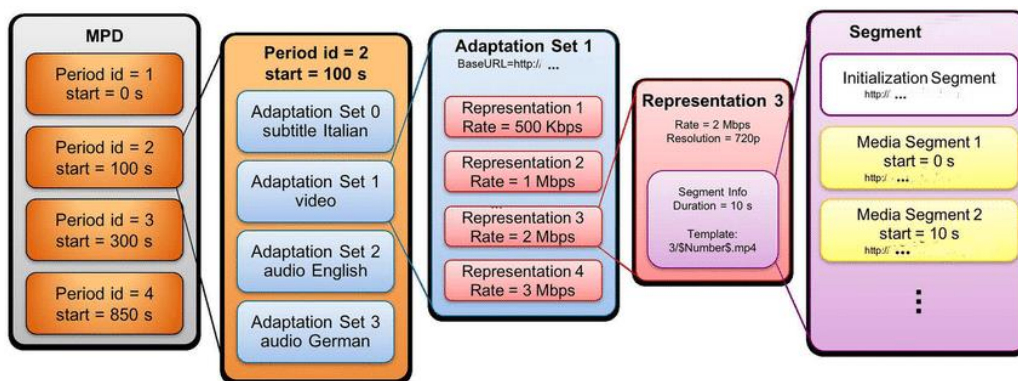


Ilustración 12 Niveles jerárquicos archivo mpd. Fuente: https://www.researchgate.net/figure/MPEG-DASH-manifest-file-structure-22_fig1_353910094

Un ejemplo de una sesión HAS contaría con los siguientes pasos (Guridi, 2022):

- 1- El cliente pide al servidor el *manifest* mediante un HTTP GET.
- 2- Del *manifest* lee los *adaptation sets* y *representations* disponibles, entre otro tipo de información.
- 3- Se seleccionan los *adaptation sets* a descargar en función del criterio establecido por el cliente.
- 4- Según algún criterio establecido se selecciona la primer *representation* a descargar de cada *adaptation set* (por ejemplo, la de menor bitrate).
- 5- Se descarga el primer segmento de las *representations* seleccionadas.
- 6- A partir de algunos parámetros como el ancho de banda o el estado del *buffer* se selecciona la siguiente *representation* a descargar de cada *adaptation set*.
- 7- Se descarga el siguiente segmento de las *representations* seleccionadas.
- 8- Se vuelve al paso 6 hasta el fin de la sesión.

5. Sistemas mulsemedia

5.1 Entornos multisensoriales

En los recientes años son cada vez más actuales las experiencias multisensoriales en la interacción humano-máquina. Estas experiencias multisensoriales se pueden implementar de forma que se estimulen diversos sentidos al mismo tiempo. Dependiendo de los sentidos que se quieran estimular se pueden utilizar diferentes combinaciones de dispositivos tecnológicos (Saleme, Covaci, Mesfin, Santos, & Ghinea, 2019):

- Dispositivos hápticos, como ratones hápticos y asientos hápticos que estimulan el sentido del tacto a través de aspectos como la retroalimentación de fuerza o la tecnología táctil.
- Dispositivos gustativos los cuales estimulan el sentido del gusto. Sin embargo, a día de hoy todavía no son muy comunes.
- Dispositivos olfativos que estimulan el sentido del olfato mediante olores. Junto a los hápticos son los más comunes en la actualidad.
- Dispositivos hechos a medida, los cuales estimulan diferentes combinaciones de sentidos y forman un entorno multisensorial único.

Un buen ejemplo de generación de efectos multisensoriales es la máscara *Feelreal VR* (Rodríguez, 2019) (ver ilustración 13), que tiene como objetivo ofrecer un mayor nivel de inmersión al usuario en su experiencia de VR. Esta máscara cuenta con un sistema de nueve esencias que se pueden elegir de entre 255 aromas disponibles. Además, también tiene unos pequeños calentadores y refrigeradores que expulsan corrientes de aire calientes y frías, respectivamente. También puede proporcionar la sensación de humedad a través de un generador de aire ionizado, e incluso cuenta con motores hápticos que pueden generar pequeñas vibraciones en la cara del usuario. Sin embargo, finalmente este proyecto nunca llegó a comercializarse, por lo que solo se quedó en una idea.

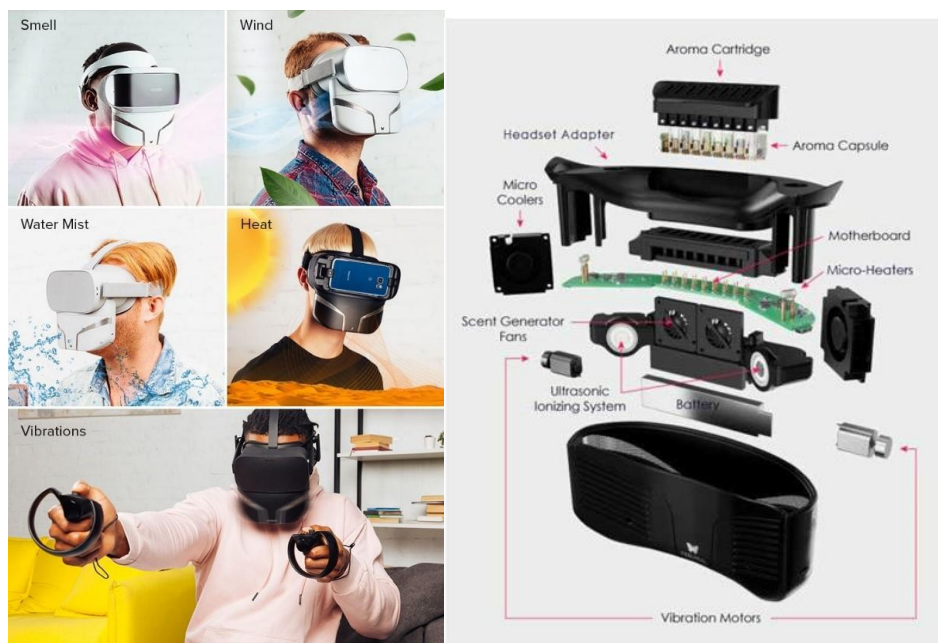


Ilustración 13 Máscara Feelreal. Fuente: <https://ryanschultz.com/2019/04/15/feelreal-multisensory-vr-mask-raises-us100000-in-kickstarter/> (imagen izquierda) y <https://www.inceptivemind.com/feelreal-add-sense-smell-vr-world/7178/> (imagen derecha)

5.1.1 Construcción de un entorno mulsemmedia

En la ilustración 14 se muestra un ejemplo de un diseño concreto de un entorno mulsemmedia tomado de la referencia (Saleme, Covaci, Mesfin, Santos, & Ghinea, 2019), el cual puede adaptarse en función de los sentidos que se quiera estimular. El elemento principal del sistema consiste en un ordenador el cual tiene una aplicación multimedia que puede reproducir contenido audiovisual y metadatos de efectos sensoriales (SEM). Dicho ordenador debe o bien, estar conectado a la red para poder descargar el contenido, o bien, tener el contenido almacenado en el disco local. Además, debe estar conectado a unos altavoces y comunicarse con el renderizador mulsemmedia, el cual será el encargado de generar los distintos efectos sensoriales.

En este caso, el renderizador mulsemmedia se comunicará con dispositivos para producir diferentes efectos sensoriales, como son un dispositivo de iluminación, uno de generación de viento, uno de generación de olores y, por último, uno de generación de vibración. Los dos primeros dispositivos están conectados vía cable a un microcontrolador, responsable del control de los dispositivos que, a su vez, está vinculado al renderizador mulsemmedia a través de un USB. Por último, el dispositivo de olfato se comunica directamente por USB con el renderizador, mientras que el dispositivo de vibración lo hace a través de la tecnología Bluetooth (Estevao B. Saleme, 2019).

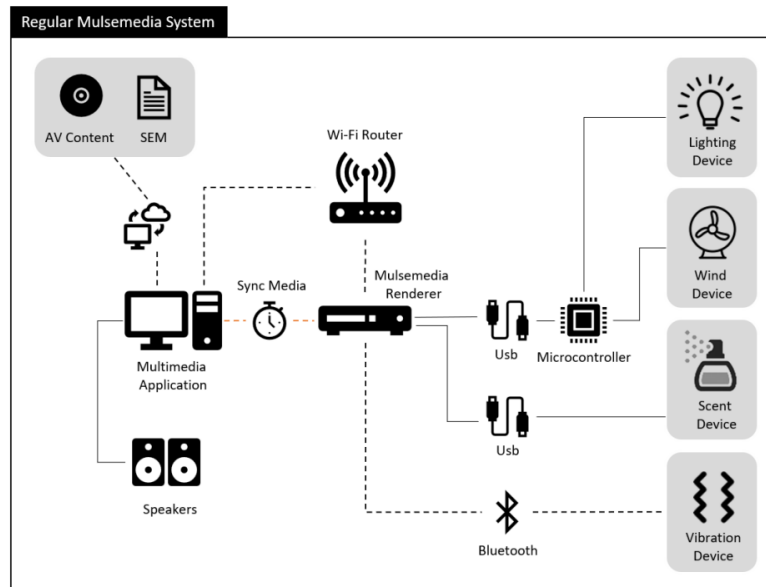


Ilustración 14 Diseño genérico de un sistema multimedial. Fuente <https://dl.acm.org/doi/pdf/10.1145/3319853>

5.2 Estandarización. MPEG-V

En MPEG-V Parte 3 (MPEG-V Sensory Information, 2009) se especifica el lenguaje de descripción de efectos sensoriales (SEDL), que es un lenguaje basado en XML para describir dichos efectos, como, por ejemplo, efectos de luz, viento, rocío, vibración, etc. Los efectos sensoriales reales no forman parte del SEDL, pero están definidos en el Vocabulario de efectos sensoriales (SEV) como extensibles y flexibles, lo que permite que cada aplicación aplique y defina sus propios efectos sensoriales.

Una descripción compatible con SEDL se denomina metadatos de efectos sensoriales (SEM) y se puede asociar con cualquier tipo de contenido multimedia (por ejemplo, películas, música, páginas web o juegos). SEM se utiliza para indicar los efectos sensoriales a los dispositivos sensoriales como ventiladores, asientos que vibran, luces, etc.

Eso significa que, por ejemplo, además del contenido audiovisual de una película, el usuario también experimentará otros efectos como los descritos anteriormente, lo que le dará la sensación de ser parte de la película, otorgándole una valiosa experiencia de información.

El contenido multimedia y SEM se puede obtener a través de DVD, disco Blu-Ray o cualquier tipo de servicio en línea. La herramienta de procesamiento de medios se denomina herramienta de RoSE, actúa como intermediario y es responsable de

reproducir los videos, que actúan como recurso, y activar de forma síncrona los efectos sensoriales que los acompañan, según la configuración del usuario. Por lo tanto, la herramienta de procesamiento de medios puede ajustar tanto el contenido de los medios como los metadatos de los efectos sensoriales, de acuerdo con las capacidades de los diferentes dispositivos de reproducción (MPEG-V Sensory Information, 2009).

El vocabulario de efectos sensoriales (SEV) define un conjunto claro de efectos sensoriales realistas para usar con el lenguaje de descripción de efectos sensoriales (SEDL) de una manera escalable y flexible. Es decir, se puede ampliar fácilmente con nuevos efectos o mediante la extracción de efectos existentes mediante la extensión XML.

Los elementos de metadatos sensoriales o los tipos de datos se asignan a comandos que controlan los dispositivos generadores de efectos en función de sus capacidades. Este mapeo normalmente lo proporciona la herramienta RoSE y no se define intencionadamente en este estándar, es decir, está abierto a la competencia de la industria. Es importante tener en cuenta que no existe necesariamente una correspondencia uno a uno entre los elementos de metadatos de efectos sensoriales o el tipo de datos y las capacidades del generador de efectos (MPEG-V Sensory Information, 2009).

5.3 Estructura del SEM

El SEM es un elemento raíz que puede contener un atributo de extracción automática opcional junto con una descripción de metadatos seguida de las opciones de declaración de elemento, grupo de efectos, efecto y efecto de referencia (Figura 3). El atributo *autoExtraction* se utiliza para indicar si se extraen automáticamente los efectos sensoriales del contenido multimedia. Los metadatos descriptivos brindan información sobre el propio SEM y alias para los esquemas de clasificación utilizados a lo largo de la descripción. Por lo tanto, se utiliza el esquema de descriptores MPEG-7.

El SEM está formado por los siguientes cinco elementos (Timmerer, Gelissen, Walti, & Hellwagner):

- Declaración: Se utiliza para definir un conjunto de elementos SEDL, sin inicializarlos, para su posterior uso en el SEM a través de una referencia interna. Este parámetro se usa para establecer parámetros comunes para varios

sensores de efectos similares a las variables en un lenguaje de programación (ver ilustración 15).

```
Declarations ::= (GroupOfEffects|Effect|
                 Parameter)+
```

Ilustración 15 Estructura declaración. Fuente: <https://www.mpeg.org/wp-content/uploads/whitepapers/files/MPEG-V/MPEG-V.pdf>

- Grupo de efectos: Comienza con una marca de tiempo que proporciona información sobre cuándo estará disponible este grupo de efectos para la aplicación. Esta información se puede utilizar con fines de visualización y sincronización con el contenido multimedia asociado. Además, un GroupOfEffects contendrá al menos dos definiciones de efectos sin marcas de tiempo necesarias porque se proporcionan en el elemento envolvente (ver ilustraciones 16 y 17).

```
GroupOfEffects ::=
timestamp EffectDefinition
EffectDefinition (EffectDefinition)*
```

Ilustración 16 Estructura grupo de efectos. Fuente: <https://www.mpeg.org/wp-content/uploads/whitepapers/files/MPEG-V/MPEG-V.pdf>

```
<sedl:GroupOfEffects si:pts="324000"
  duration="100" fade-in="15" fade-out="15"
  position="urn:mpeg:mpeg-v:01-SI-
  PositionCS-NS:center:*:front">
  <sedl:Effect xsi:type="sev:WindType"
    intensity="0.0769"/>
  <sedl:Effect
    xsi:type="sev:TemperatureType"
    intensity="0.777"/>
  <sedl:Effect xsi:type="sev:LightType"
    intensity="0.0000077"/>
</sedl:GroupOfEffects>
```

Ilustración 17 Ejemplo grupo de efectos. Fuente: <https://www.mpeg.org/wp-content/uploads/whitepapers/files/MPEG-V/MPEG-V.pdf>

- Efecto: Se utiliza para describir un solo efecto con una marca de tiempo asociada (ver ilustraciones 18 y 19).

```
Effect ::= timestamp EffectDefinition
```

Ilustración 18 Estructura efecto. Fuente: <https://www.mpeg.org/wp-content/uploads/whitepapers/files/MPEG-V/MPEG-V.pdf>

```
<sedl:Effect xsi:type="sev:VibrationType"  
  intensity="0.56" duration="..."  
  si:pts="..." .../>
```

Ilustración 19 Ejemplo de efecto. Fuente: <https://www.mpeg.org/wp-content/uploads/whitepapers/files/MPEG-V/MPEG-V.pdf>

- Efecto de definición: Tiene un elemento de información suplementaria para definir un área de referencia desde la cual se puede extraer información del efecto en caso de que la extracción automática esté habilitada (ver ilustración 20). Además, define varias propiedades opcionales:
- *Atributo de activación*: Como sugiere su nombre, su función es ser responsable de desencadenar el efecto sensorial
 - *Atributo de duración*: Indica la duración del efecto sensorial.
 - *Atributo de desaparición gradual*: Especifica el tiempo de atenuación del efecto sensorial.
 - *Atributo de aparición gradual*: Especifica el tiempo en el cuál aparece el efecto sensorial.
 - *Atributo alt*: Permite especificar un efecto de reemplazo en caso de que no se pueda crear el efecto original por falta de recursos o dispositivos.
 - *Atributo de prioridad*: Indica la prioridad entre los diferentes efectos que comparten el mismo momento en el que deberían estar disponibles para el consumo. Un valor de 1 indica la prioridad más alta y un valor más alto indica una prioridad más baja.
 - *Atributo de intensidad*: Indica la intensidad del efecto en porcentaje según una escala o unidad predefinida.

- *Atributo de posición:* Especifica la ubicación desde donde se espera que se reciba el efecto.

```
EffectDefinition ::= [SupplementalInformation]
[activate] [duration] [fade-in] [fade-out]
[alt] [priority] [intensity] [position]
[adaptability] [autoExtraction]
```

Ilustración 20 Estructura efecto de definición. Fuente: <https://www.mpeg.org/wp-content/uploads/whitepapers/files/MPEG-V/MPEG-V.pdf>

El efecto de definición también incluye un grupo de atributos de adaptabilidad que permiten la descripción del tipo de adaptación preferido con un límite superior e inferior determinados.

6. Grabación del material

6.1 Proceso de preproducción

6.1.1 Localizaciones del proyecto

Al principio del proyecto, se decide qué lugares y localizaciones deben ser incluidas en el proyecto. Teniendo en cuenta que se trata de un proyecto destinado a la Cátedra de Gandía Turismo Inteligente del Campus de Gandía de la UPV, se ha decidido realizar la grabación del material en cuatro de los lugares más icónicos de Gandía y sus alrededores, como son la Cueva del Parpalló, la Marjal, la Alquería del Duc y la propia playa de Gandía, concretamente en el paseo marítimo de esta. Estos cuatro lugares han sido los escogidos, ya que son los cuatro lugares con los que se ha trabajado anteriormente desde la Cátedra y son los que forman parte de su página web.

Una vez claras las localizaciones que iban a ser incluidas en el proyecto, se realiza un desplazamiento para conocer mejor la zona y reconocer cuáles serían los lugares óptimos para el emplazamiento de la cámara. Este desplazamiento a las localizaciones, además de acortar tiempos el día de grabación, facilita el proceso de realización tanto del guion técnico como del plan de rodaje.

6.1.2 Guion técnico

Después de realizar la elección de las localizaciones del proyecto, se procede a hacer un guion técnico para tener una base sobre la que primero grabar y, posteriormente,

editar, teniendo en cuenta los tiempos de cada clip. Antes de realizar el guion técnico, se decide que en cada localización se utilizarán cinco vídeos, correspondientes con cinco posiciones de cámara, excepto en la Alquería del Duc, localización de la cual solo se tienen cuatro posiciones de cámara distintas. Esto se debe a que, al ser un sitio más pequeño, se ha decidido que se requerían menos vídeos para mostrar dicha localización.

Por otro lado, se decide que la coherencia narrativa del resultado final se va a obtener a partir de realizar, a través de un mapa, el recorrido en bicicleta de una localización a otra. Con esto se consigue que el resultado final se perciba como un recorrido turístico por algunos de los lugares más relevantes de la localidad de Gandía. La idea parece de lo representado en la ilustración 21, pero limitando el mapa a la zona de Gandía y alrededores y con una bicicleta como vehículo. Se elige que el vehículo utilizado sea una bicicleta debido a que las distancias que se recorren no son muy extensas, por lo que no es necesario realizar los trayectos en coche. Además, con el uso de la bicicleta, se pretende fomentar el ejercicio físico, el cuidado del medio ambiente y el uso del transporte sostenible.



Ilustración 21 Idea de cómo realizar los recorridos entre ubicaciones. Fuente: https://www.freepik.es/vector-premium/viajes-mundiales-mapas-aviones_2810052.html

Por otro lado, como secuencia inicial, se decide crear una animación con *Google Earth*, como se explicará posteriormente, que consiste en realizar un *zoom in* desde el espacio, desde donde se ve todo el planeta Tierra, hasta la zona del mapa que abarca las cuatro localizaciones. Con esta animación, lo que se pretende es situar al espectador al inicio del vídeo con tal de que tenga claro cuál es la zona geográfica que va a ver.

Una vez se tiene claro cómo se va a otorgar de coherencia narrativa al vídeo, se decide el orden de los planos para poder realizar el guion técnico correctamente. Para ello, se

ubican las cuatro localizaciones en un mapa, con ayuda del *software Google Earth Pro*, para decidir cuál es el recorrido óptimo (ver ilustración 22).

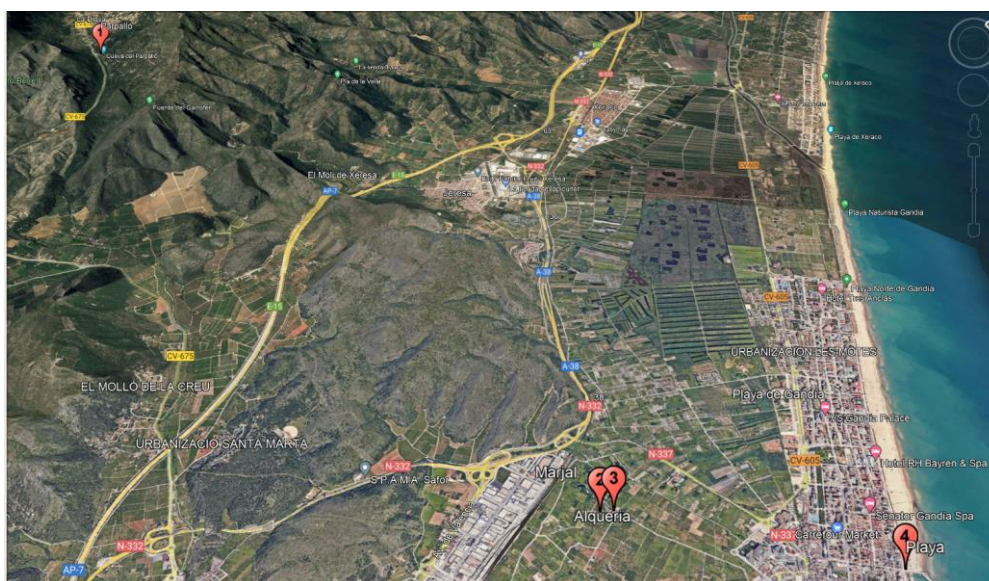


Ilustración 22 Las cuatro localizaciones ubicadas en un mapa. Elaboración propia

Con esto, se decide que el orden óptimo será de izquierda a derecha, de la montaña a la playa. Es decir, el orden elegido es “Parpalló”, “Marjal”, “Alquería” y, finalmente, “Playa”. Y, por tanto, el guion técnico queda tal y como se muestra en la ilustración 23.

LOCALIZACIÓN	ESCENA	PLANO	ACCIÓN	DURACIÓN	DURACIÓN TOTAL
Mapa	1	1	Zoom in en un mapa hacia Gandía	20"	20"
Mapa	1	2	Zoom in en Parpalló	5"	25"
Parpalló	2	3	Video 1	20"	45"
Parpalló	2	4	Video 2	20"	1' 05"
Parpalló	2	5	Video 3	20"	1' 25"
Parpalló	2	6	Video 4	20"	1' 45"
Parpalló	2	7	Video 5	20"	2' 05"
Mapa	3	8	Traslado en bici desde Parpalló a Marjal	5"	2' 10"
Marjal	4	9	Video 1	20"	2' 30"
Marjal	4	10	Video 2	20"	2' 50"
Marjal	4	11	Video 3	20"	3' 10"
Marjal	4	12	Video 4	20"	3' 30"
Marjal	4	13	Video 5	20"	3' 50"
Mapa	5	14	Traslado en bici desde Marjal a Alquería	5"	3' 55"
Alquería	6	15	Video 1	20"	4' 15"
Alquería	6	16	Video 2	20"	4' 35"
Alquería	6	17	Video 3	20"	4' 55"
Alquería	6	18	Video 4	20"	5' 15"
Mapa	7	19	Traslado en bici desde Alquería a Playa	5"	5' 10"
Playa	8	20	Video 1	20"	5' 30"
Playa	8	21	Video 2	20"	5' 50"
Playa	8	22	Video 3	20"	6' 10"
Playa	8	23	Video 4	20"	6' 30"
Playa	8	24	Video 5	20"	6' 50"

Ilustración 23 Guion técnico. Elaboración propia

Es importante remarcar que el guion técnico tan solo se ha realizado sobre el vídeo que incluye las cuatro localizaciones. Esto se debe a que el guion técnico de los vídeos individuales de cada localización son un extracto del guion técnico del vídeo principal, ya que el video principal incluye las cuatro localizaciones y, por tanto, su guion técnico incluye el guion técnico de las cuatro localizaciones. Extrayendo dichos guiones técnicos quedarían tal y como se muestran en las ilustraciones 24, 25, 26 y 27.

LOCALIZACIÓN	ESCENA	PLANO	ACCIÓN	DURACIÓN	DURACIÓN TOTAL
Parpalo	1	1	Video 1	20"	20"
Parpalo	1	2	Video 2	20"	40"
Parpalo	1	3	Video 3	20"	1'
Parpalo	1	4	Video 4	20"	1' 20"
Parpalo	1	5	Video 5	20"	1' 40"

Ilustración 24 Guion técnico "Parpalló". Elaboración propia

LOCALIZACIÓN	ESCENA	PLANO	ACCIÓN	DURACIÓN	DURACIÓN TOTAL
Marjal	4	9	Video 1	20"	20"
Marjal	4	10	Video 2	20"	40"
Marjal	4	11	Video 3	20"	1'
Marjal	4	12	Video 4	20"	1' 20"
Marjal	4	13	Video 5	20"	1' 40"

Ilustración 25 Guion técnico "Marjal". Elaboración propia

LOCALIZACIÓN	ESCENA	PLANO	ACCIÓN	DURACIÓN	DURACIÓN TOTAL
Alqueria	6	15	Video 1	20"	20"
Alqueria	6	16	Video 2	20"	40"
Alqueria	6	17	Video 3	20"	1'
Alqueria	6	18	Video 4	20"	1' 20"

Ilustración 26 Guion técnico "Alquería". Elaboración propia

LOCALIZACIÓN	ESCENA	PLANO	ACCIÓN	DURACIÓN	DURACIÓN TOTAL
Playa	8	20	Video 1	20"	20"
Playa	8	21	Video 2	20"	40"
Playa	8	22	Video 3	20"	1'
Playa	8	23	Video 4	20"	1' 20"
Playa	8	24	Video 5	20"	1' 40"

Ilustración 27 Guion técnico "Playa". Elaboración propia

La duración de los distintos vídeos que forman parte de las cuatro localizaciones se ha fijado en 20 segundos, con el objetivo de no obtener unos vídeos demasiado largos, lo cual, aumentaría el espacio que ocupan (gigabytes) y, por tanto, el tiempo de trabajo con las distintas herramientas y *softwares* por las que pasarán los vídeos posteriormente. Mientras que la duración de los traslados en bicicleta se ha estimado que tienen una duración media de 5 segundos. No obstante, está duración puede variar de un traslado a otro dependiendo de la distancia del recorrido. Ocurriendo lo mismo con la duración de la animación inicial, pues se ha realizado una estimación y, posteriormente, al realizar la animación se ajustará mejor. Esto se debe a que

umentando o reduciendo la duración se controla el ritmo de la animación, por lo que hasta no realizar la animación no se puede fijar con exactitud su duración.

6.1.3 Plan de rodaje

Del mismo modo que se ha elaborado un guion técnico, también se requiere elaborar un plan de rodaje (ver ilustración 28), el cual recoge un desglose, por días, de cada una de las escenas y planos que van a ser grabados. Como se puede observar, el plan de rodaje se programó para dos días, donde en el primer día, por la tarde, se graba en la localización de la playa, mientras que, en el segundo día se utiliza la mañana para grabar en las otras tres localizaciones. Del plan de rodaje también se puede observar que se dedican 10 minutos a la grabación de cada vídeo, lo cual permite asegurarse de que el emplazamiento de cámara es el correcto y de que la grabación se ha realizado correctamente.

Día 1		14 DE JUNIO DE 2023			
Nº	Hora	Plano	Cámara	Localización	Audio
1	17:30	20	Cámara fija	Playa	Música
2	17:40	21	Cámara fija	Playa	Música
3	17:50	22	Cámara fija	Playa	Música
4	18:00	23	Cámara fija	Playa	Música
5	18:10	24	Cámara fija	Playa	Música
Día 2		15 DE JUNIO DE 2023			
Nº	Hora	Plano	Cámara	Localización	Audio
1	11:00	1	Cámara fija	Parpalló	Música
2	11:10	2	Cámara fija	Parpalló	Música
3	11:20	3	Cámara fija	Parpalló	Música
4	11:30	4	Cámara fija	Parpalló	Música
5	11:40	5	Cámara fija	Parpalló	Música
1	12:00	15	Cámara fija	Alqueria	Música
2	12:10	16	Cámara fija	Alqueria	Música
3	12:20	17	Cámara fija	Alqueria	Música
4	12:30	18	Cámara fija	Alqueria	Música
1	12:40	9	Cámara fija	Marjal	Música
2	12:50	10	Cámara fija	Marjal	Música
3	13:00	11	Cámara fija	Marjal	Música
4	13:10	12	Cámara fija	Marjal	Música
5	13:20	13	Cámara fija	Marjal	Música

Ilustración 28 Plan de rodaje

6.1.4 Cámara utilizada. Características y *software*

6.1.1 Cámara utilizada

El material se ha grabado con la cámara Qoocam 8k Enterprise, la cual permite grabar videos 360° a una resolución de 8k (7680 x 3840) a 30 fps o una resolución de 4k (3840*1920) a 120fps (ver ilustración 29), siendo la primera opción la empleada para la grabación de dicho material. Esta decisión viene por la elección de priorizar la resolución

por encima de los fps, ya que posteriormente, a lo largo del proyecto, se van a crear copias del contenido a menor calidad, por lo que se prefiere partir de la máxima calidad posible. Además, esta cámara también tiene la posibilidad de transmitir contenido en vivo (ver ilustración 30).

Lens	F2.0, FOV: 200°
Image sensor size	1/1.7 inch BSI-COMS 20MP
Video (360)	7680*3840@30fps, 8/10bit 3840*1920@120fps, 8/10bit
Photo (360)	7680*3840, 12bit DNG(RAW),DNG8(16bit output)
Video format	MP4
Audio format	PCM
Photo format	JPG, DNG
Video bit rate	Up to 200Mbps
Video code	H.264, H.265
IMU	6-axis IMU
Screen	2.4 inch colorful touch screen

Ilustración 29. Especificaciones de la cámara Qoocam 8k Enterprise. Fuente: <https://www.kandaovr.com/qoocam-8k-enterprise/>

Live

Live Mode	Cam streaming , phone streaming
Supporting Protocol	rtsp, rtmp, rtmps
Resolution	8K: 7680*3840 4K: 3840*1920
Lens Adjustments	Real-time automatic/manual exposure and white balance adjustments: EV, Shutter, ISO, Color temperature
Live Preview	Camera real-time 360 preview
Live Bitrate	Max 150Mb/s
Code Format	H.264, H.265
Image Stitching	Real-time stitching
Image Stabilization	SuperSteady

Ilustración 30. Especificaciones de la cámara en la retransmisión en vivo. Fuente: <https://www.kandaovr.com/qoocam-8k-enterprise/>

6.1.2 Software

En la propia web de Kandao⁴, empresa a la cual pertenece la cámara utilizada, se encuentra una serie de *software* asociados a todos los modelos de cámara que comercializan. De todos los asociados a la Qoocam 8k Enterprise, para la realización de este proyecto se utilizan dos de estos programas.

En primer lugar, se descarga y utiliza la aplicación QooCam App, la cual está disponible tanto para Android como para iOS. Qoocam App permite la grabación y reproducción de contenido desde el móvil además de poder cambiar la configuración de la cámara y acceder a una serie de tutoriales (ver ilustración 31). El uso de esta aplicación supone un mejor manejo de la cámara y la opción de poder controlar la grabación aun estando a cierta distancia de la cámara, teniendo en cuenta las facilidades que esto proporciona.



Ilustración 31 Tutoriales app QooCam App. Elaboración propia. Captura de pantalla QooCam App

En segundo lugar, se descarga y utiliza el *software* QooCam Studio 2.0, el cual es indispensable para la exportación correcta de los videos, ya que si se obtienen los videos directamente de la cámara no se pueden reproducir debido a que no se incluyen los metadatos de este.

⁴ <https://www.kandaovr.com/download/>

6.2 Grabación del material

Siguiendo el plan de rodaje y el guion técnico adjuntados en los apartados anteriores, se realiza la grabación del material. En cuanto al emplazamiento de cámara, tal y como se propuso desde un principio, se ha procurado realizarlo de forma que las distintas ubicaciones de la cámara cubrieran toda la localización.

Así pues, con dicho material se van a realizar cinco vídeos, uno por cada localización y otro que incluya las cuatro localizaciones, teniendo así un vídeo final que dote al proyecto de coherencia narrativa, siendo el proceso de edición de este último el que se ha explicado más detenidamente a lo largo del documento.

6.3 Exportación del material

A medida que el material es grabado, se pasan los archivos de la cámara al ordenador, ordenados según su ubicación (ver ilustración 32). Para ello se requiere conectar la cámara al ordenador mediante un cable de USB tipo C a USB 3.0.

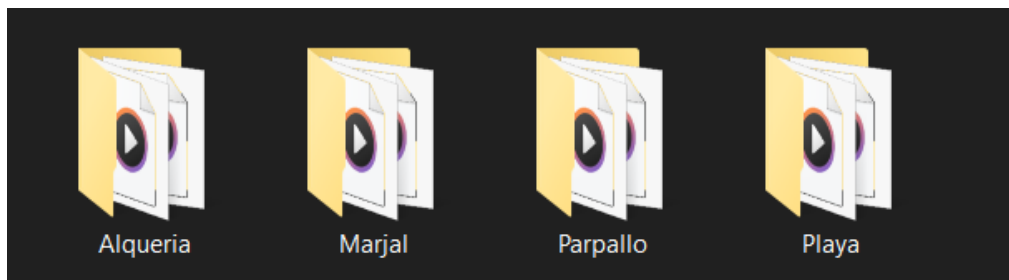


Ilustración 32. Organización de los brutos. Elaboración propia

Sin embargo, sucede que estos vídeos al extraerlos de la cámara no se pueden reproducir en 360° directamente (ver ilustración 33). Para que esto sea posible, se deben pasar por el *software* QooCam Studio 2.0 mencionado anteriormente (ver ilustración 34).



Ilustración 33. Preview del vídeo sin el software QooCam Studio 2.0. Elaboración propia

Por tanto, se importan los vídeos en dicho *software*. Este programa ofrece como ajustes “*yaw, pitch y roll*” (ver ilustración 35), los cuales pueden tener un valor entre -180° y 180° . Mediante estos ajustes, el *software* permite alinear la línea del horizonte correctamente y que el resultado final del vídeo no cuente con algún tipo de inclinación no deseada. También el propio *software* cuenta con la opción de activar una rejilla con tal de facilitar este ajuste de la inclinación del vídeo acorde a la línea del horizonte.

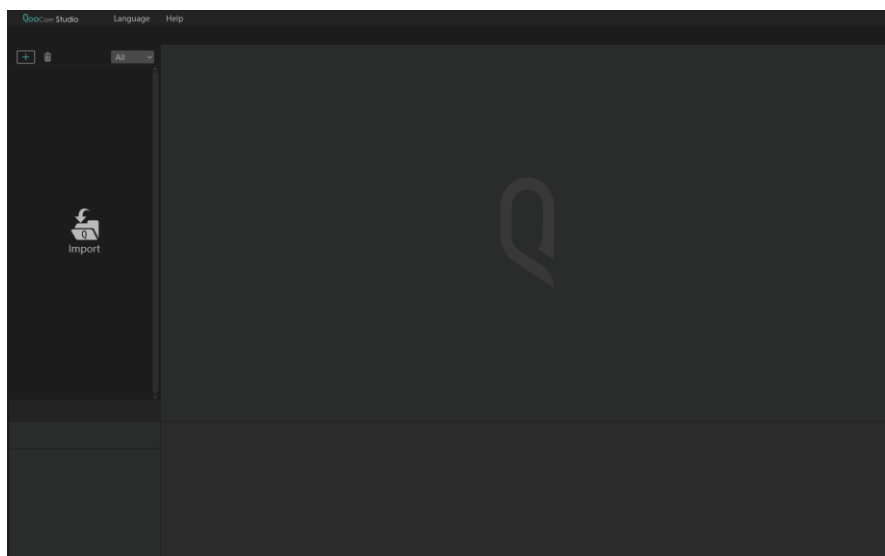


Ilustración 34. Pantalla de inicio del QooCam Studio 2.0. Elaboración propia

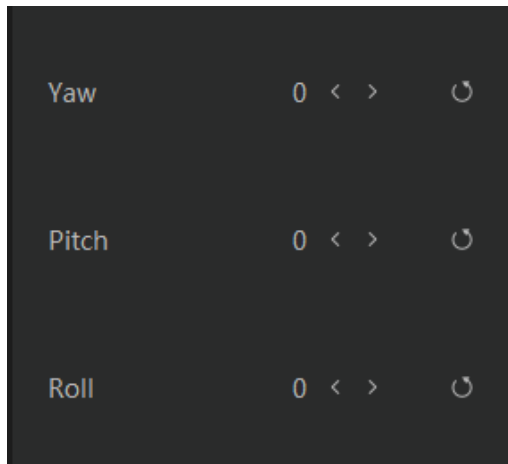


Ilustración 35 Ajustes en el software QooCam Studio. Elaboración propia

Finalmente, se tiene otro menú de opciones mediante el cual se puede controlar la estabilización del vídeo, el número de fotogramas a los que se va a exportar el vídeo o, incluso, aplicar algunas correcciones de color (ver ilustración 36).

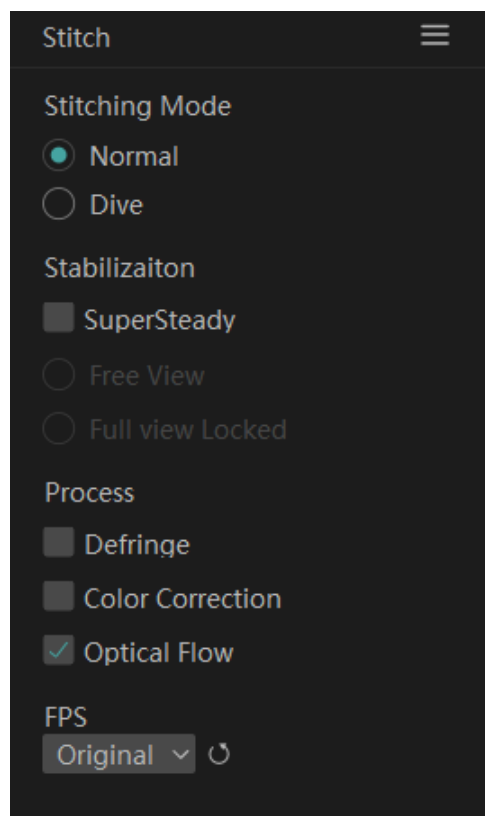


Ilustración 36 Menú de opciones de Stitch. Elaboración propia

En ningún vídeo se realizan ajustes mediante este menú debido a que no se necesita estabilización al grabar desde una posición fija con un trípode y, además, se ha considerado que no es necesario aplicar la corrección de color. Esta decisión viene dada por el hecho de que posteriormente, en la edición de los vídeos ya se va a realizar un

proceso de etalonaje y corrección de color. Una vez ajustados estos parámetros, se procede a la exportación de los vídeos para su posterior organización y edición.

7. Proceso de posproducción

7.1 Organización del material

Así pues, la organización que se utiliza es, en primer lugar, dividir los videos al exportarlos según la localización a la que pertenezcan (Parpalló, Marjal, Alquería del Duc o Playa) (ver ilustración 37).

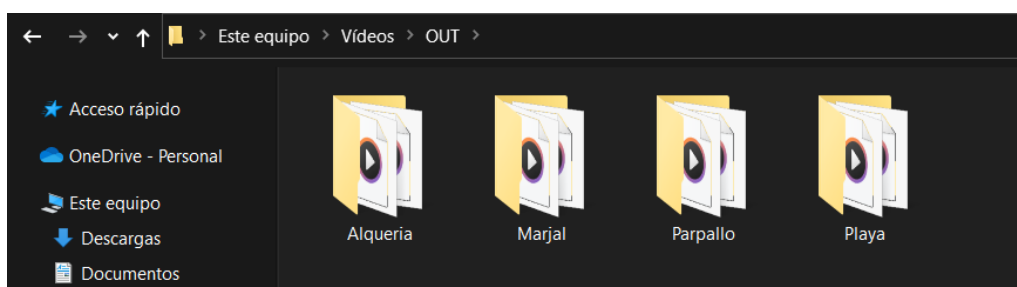


Ilustración 37 Primer nivel de organización del material. Elaboración propia

Y dentro de estas carpetas se renombran los vídeos según la posición de cámara a la que correspondan, catalogando cada posición de cámara como una ubicación (ver ilustración 38).

Nombre	Fecha	Tipo	Tamaño	Duración
Ubicación_1.mp4	19/06/2023 19:38	Archivo MP4	411.331 KB	00:00:26
Ubicación_2.mp4	19/06/2023 19:37	Archivo MP4	615.160 KB	00:00:40
Ubicación_3.mp4	19/06/2023 19:34	Archivo MP4	1.059.688 ...	00:00:40
Ubicación_4.mp4	19/06/2023 19:36	Archivo MP4	1.144.615 ...	00:00:40
Ubicación_5.mp4	19/06/2023 19:32	Archivo MP4	1.086.110 ...	00:00:40
Ubicación_6.mp4	19/06/2023 19:33	Archivo MP4	702.112 KB	00:00:51
Ubicación_7.mp4	19/06/2023 19:30	Archivo MP4	1.205.981 ...	00:00:50
Ubicación_8.mp4	19/06/2023 19:29	Archivo MP4	799.367 KB	00:00:45

Ilustración 38 Segundo nivel de organización del material. Elaboración propia

7.1.1 Visionado del material

Una vez se tiene el material organizado, se pasa al visionado de este. Con el primer visionado se buscan dos objetivos.

- 1- Analizar el material del que se dispone y ver si hay que descartar algún vídeo ya sea por su calidad, algún fallo de grabación o porque su contenido, dada la posición de cámara, no aporta mucha información valiosa.

2- Analizar si hay algunas posiciones de cámara claves para dar el valor que se merece a cada localización. En este caso se apunta la ubicación de cámara que es imprescindible que forme parte del proyecto.

Normalmente la selección de planos se realiza buscando un balance entre el nivel técnico y el nivel artístico. Debido a las características demandadas por el proyecto se ha optado por grabar en parajes naturales, con iluminación natural y sin interpretación actoral. Por eso, en este caso la selección se realiza teniendo en cuenta que se pueda apreciar los máximos detalles de la localización con la menor cantidad de vídeos posibles, reduciendo así el tiempo del vídeo final. Esta reducción de tiempo se busca principalmente para evitar tener un vídeo muy pesado, el cual ocupe demasiados gigabytes (GB), ya que luego debe ser tratado con diversos *softwares* y el aumento de peso del vídeo está directamente relacionado con el tiempo que tardan dichos *softwares* en realizar el trabajo.

7.2 Edición del material

La edición del vídeo se divide en tres partes: en primer lugar, la animación inicial, seguida de los traslados en bicicleta y finalmente la edición de los vídeos 360°.

7.2.1 Animación inicial

La animación inicial se realiza con ayuda de la herramienta online *Google Earth Studio*, la cual tiene un funcionamiento muy intuitivo y mediante la cual se puede animar la posición y la rotación de la cámara. Así pues, mediante *keyframes* (siendo estos los puntos blancos de la ilustración 39) se van ajustando ambas propiedades para obtener el resultado deseado.

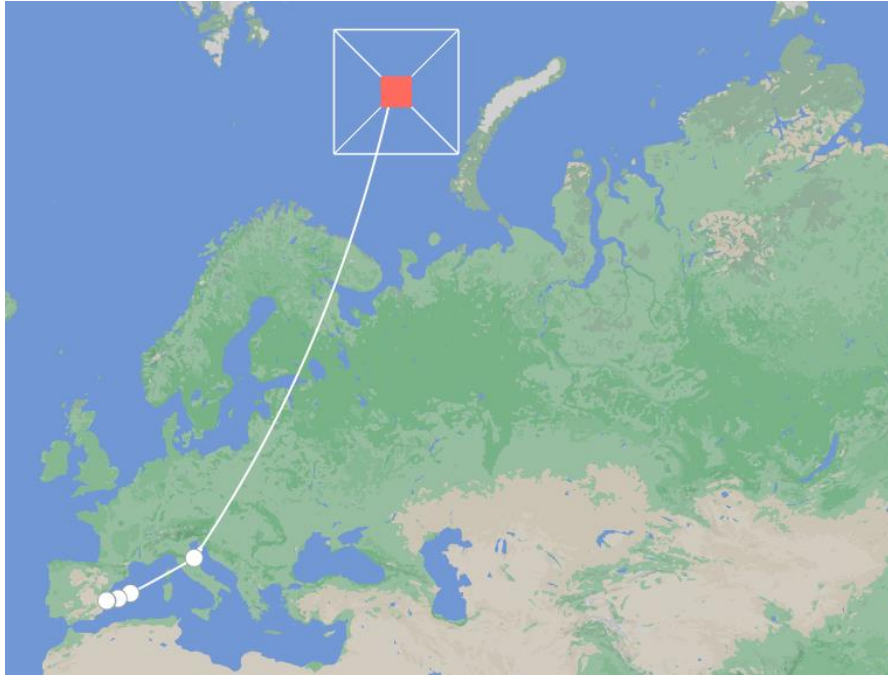


Ilustración 39 Resumen edición con Google Earth Studio. Elaboración propia

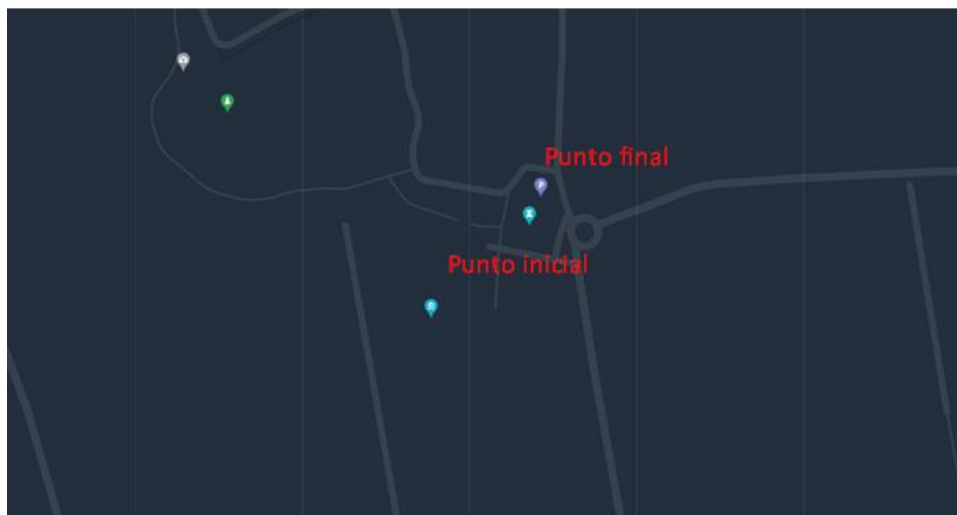
A la hora de renderizar, esta herramienta permite dos opciones: una en la que se renderiza como vídeo en formato mp4 y otra en la que se renderiza como una secuencia de imágenes en formato jpeg. La diferencia entre las dos opciones es que si se renderiza directamente como un vídeo se obliga a que el mapa sea limpio, es decir, que no haya ninguna etiqueta que dé información del lugar en el que se está. Es por esto por lo que se decide renderizar como una secuencia de imágenes para así tener etiquetas que dan información de qué países y ciudades se están viendo en ese momento. Finalmente, se juntan todas las imágenes, con una duración de 0,03 segundos, formando así una secuencia de catorce segundos.

7.2.2 Recorridos en bicicleta

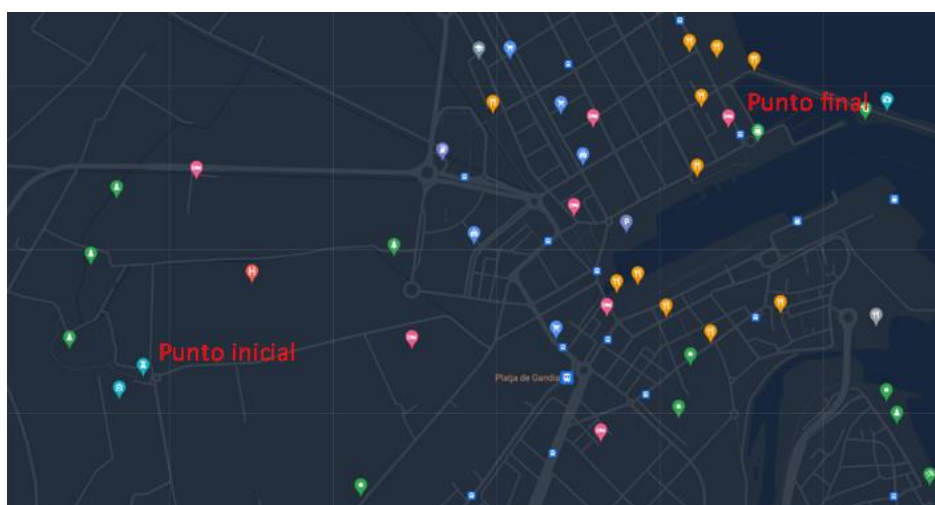
La siguiente parte de la edición, los traslados con la bicicleta, es la más compleja y elaborada. En primer lugar, se utiliza la herramienta *mapstyle* de Google para obtener una imagen de los mapas sobre los que posteriormente se va a colocar la bicicleta. Al igual que en la herramienta anterior, esta herramienta permite seleccionar el grado de detalle de las etiquetas. En este caso se utiliza un detalle mínimo, dado que se considera que un grado mayor de detalle quitaría protagonismo a la bicicleta y al recorrido, que es lo que se quiere remarcar. Además, este mapa se puede obtener con diferentes estilos y colores. Por temas de visibilidad y eficiencia se elige el tema “noche”, obteniendo los resultados mostrados en las ilustraciones 40, 41 y 42.



Il·lustració 40. Mapa recorridu Parpalló-Marjal. Elaboració pròpia



Il·lustració 41. Mapa recorridu Marjal – Alquería. Elaboració pròpia



Il·lustració 42. Mapa recorridu Alquería-Playa. Elaboració pròpia

Una vez se han obtenido los mapas, se vectorizan mediante el *software Adobe Illustrator 2023*. Vectorizando las imágenes lo que se consigue es que se puedan ampliar sin que se añada ruido a dicha imagen. Este proceso de vectorización también se aplica al archivo .jpeg de la bicicleta, mostrada en la ilustración 43.



Ilustración 43 Bicicleta. Fuente: <https://www.freepng.es/png-hmhws4/>

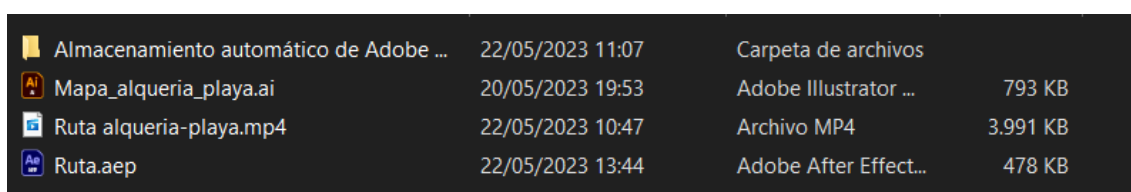
Con los archivos vectorizados, se pasa a trabajar con el *software Adobe After Effects 2023*, mediante el cual se realizan las tres animaciones correspondientes a los tres recorridos en bicicleta. Para ello se requieren tres archivos (uno por recorrido) en los que primero se importan los archivos necesarios, como el mapa y la bicicleta. A continuación, se traza el recorrido de la bicicleta y se duplica la capa y se reduce la opacidad de la capa duplicada, consiguiendo así que parezca que el recorrido se va marcando a medida que la bicicleta avanza (ver ilustración 44).



Ilustración 44 Diferencia de opacidad entre el recorrido ya realizado y el que está por realizar. Elaboración propia

Finalmente, se ajustan los movimientos de cámara para que estos acompañen al recorrido y así darle un toque más dinámico.

Después de este proceso y habiendo exportado los archivos creados con *After Effects*, se tiene que en esta parte de la edición por cada uno de los recorridos se han generado tres archivos, tal y como se muestra en la ilustración 45, los cuales contienen el mapa vectorizado, el archivo de *After Effects* y el archivo exportado en formato mp4, que es el que se usará en el resultado final.



Almacenamiento automático de Adobe ...	22/05/2023 11:07	Carpeta de archivos	
Mapa_alqueria_playa.ai	20/05/2023 19:53	Adobe Illustrator ...	793 KB
Ruta alqueria-playa.mp4	22/05/2023 10:47	Archivo MP4	3.991 KB
Ruta.aep	22/05/2023 13:44	Adobe After Effect...	478 KB

Ilustración 45. Organización material recorridos. Elaboración propia

7.2.3 Edición de videos 360°

Con esto se llega a la última parte de la edición, la edición de los vídeos 360°. Para la edición de videos 360° se planteó utilizar el software *DaVinci Resolve*, pero finalmente se utilizó el programa *Adobe Premier Pro 2023*, ya que se consideró que se adaptaba mejor a la edición de videos 360°.

En primer lugar, se importan todos los vídeos previamente seleccionados y se ordenan por carpetas dentro del programa *Adobe Premier Pro 2023*. Además, con tal de tener una mejor organización, a los videos se les asigna una etiqueta, el color de la cual depende de la localización a la que pertenecen, y también se etiqueta a qué plano y escena pertenecen (ver ilustración 46).

Nombre	li	Escena	Plano	Punto de entrada d	Punto de salida de	Duración del vídeo	Inicio del subclip
> Alqueria							
Marjal							
Ubicación 10				00:00:00:0	00:00:19:0	00:00:20:0	
Ubicación 1	4		9	00:00:00:0	00:00:18:0	00:00:19:0	
ubicación 5	4		11	00:00:00:0	00:00:19:0	00:00:20:0	
Ubicación 9	4		12	00:00:13:0	00:00:32:0	00:00:20:0	
Ubicación 12	4		12	00:00:10:0	00:00:29:0	00:00:20:0	
Parpallo							
ubicación 1	2		3	00:00:23:0	00:00:42:0	00:00:20:0	
ubicación 2	2		4	00:00:00:0	00:00:19:0	00:00:20:0	
ubicación 3	2		5	00:00:23:0	00:00:42:0	00:00:20:0	
ubicación 4	2		6	00:00:00:0	00:00:19:0	00:00:20:0	
ubicación 5	2		7	00:00:00:0	00:00:19:0	00:00:20:0	
Playa							
ubicación 1	8		20	00:00:00:0	00:00:19:0	00:00:20:0	
ubicación 3	8		21	00:00:00:0	00:00:19:0	00:00:20:0	
Ubicación 4	8		22	00:00:30:0	00:00:49:0	00:00:20:0	
ubicación 7	8		23	00:00:00:0	00:00:19:0	00:00:20:0	
ubicación 8	8		24	00:00:40:0	00:00:59:0	00:00:20:0	

Ilustración 46. Ejemplo organización del material en Adobe Premier Pro 2023. Elaboración propia

Una vez se han organizado estos archivos se crea otra carpeta llamada “rutas” que contiene las animaciones, manteniendo así el orden en los archivos importados (ver ilustración 47).

Nombre	Punto de entrada d	Punto de salida de	Duración del vídeo	Inicio del subclip
Rutas				
Animación inicial.mp4	00:00:00:00	00:00:14:24	00:00:14:25	
Ruta alqueria-playa.mp	00:00:00:00	00:00:09:24	00:00:10:00	
Ruta parpallo-marjal.m	00:00:00:00	00:00:13:24	00:00:14:00	
Ruta_marjal_alqueria.	00:00:00:00	00:00:04:24	00:00:05:00	

Ilustración 47. Organización de los recorridos en Adobe Premier Pro 2023. Elaboración propia

En cuanto al proceso de edición, en este proyecto no se cuenta con sonido ambiente por lo que no se deber realizar la sincronización entre el audio y la imagen. Este proceso permite enlazar el audio grabado con un micrófono o con una grabadora con la imagen, la cual suele ir acompañada con el sonido captado por la propia cámara. Sin embargo, en esta ocasión no se tiene nada de audio, ni captado por la propia cámara ni por un micrófono exterior. Por lo tanto, se ahorra el tiempo dedicado a este proceso y se añadirá, posteriormente, una canción que acompañe al vídeo. Así pues, lo primero que se hace en este proyecto es revisar los clips uno a uno, marcando un punto de inicio y un punto de fin, seleccionando así veinte segundos de cada clip.

Cuando ya se tienen todos los clips reducidos a veinte segundos, se procede a colocarlos en el *timeline* con el orden establecido en el guion técnico para, a continuación, colocar transiciones entre los clips. Se decide añadir transiciones entre

los clips en lugar de utilizar un corte porque se considera que así es menos abrupto el cambio de emplazamiento de cámara, siendo más "suave" para el usuario, ya que entre cada clip hay un salto espacial al cambiar el emplazamiento de cámara.

Terminado el proceso de edición de los videos 360°, se añaden los clips de la animación inicial y los recorridos en bicicleta, añadiendo también transiciones entre los clips. Al exportar esta primera versión se encuentra un problema y es que, al combinar vídeos 2D con vídeos 360°, *Premier Pro* asume que todo son vídeos 360° y, por tanto, de los vídeos que no lo son, convierte automáticamente su proyección a una esfera. Para evitar esto, se aplica un efecto llamado "de esfera a plano" que consiste en convertir la proyección en esfera en una proyección en un plano, tal y como lo era en un principio.

No obstante, los problemas de combinar videos 2D con vídeos 360° no se limitan a lo anteriormente nombrado. En un vídeo 360°, el usuario tiene la posibilidad de rotar la imagen arriba y abajo o derecha e izquierda, mientras que en un vídeo 2D la proyección es fija en un plano. Al combinar ambas opciones en un solo producto final lo que se tiene es que existe la posibilidad de perder el foco del vídeo 2D, ya que este tan solo se reproduce en un plano. Concretamente, se reproduce en la dirección que miraría el usuario si no moviera el foco en ningún momento, es decir, se reproduce enfrente del usuario. Este es un problema que no tiene solución ya que *Adobe Premier Pro* no permite fijar en una dirección el punto de vista del usuario.

7.3 Tratamiento del sonido

Como ya se ha comentado anteriormente, este proyecto no cuenta con sonido procedente de la propia cámara o un micrófono exterior. Es por ello por lo que, una vez se termina con el proceso de edición, se procede con la búsqueda de la música. La música solo requiere cumplir con dos requisitos, que sea libre de derechos de autor y que pueda ser asociada fácilmente con un vídeo turístico, es decir, que no desentone con lo que se está visualizando.

Tras realizar la búsqueda y tras probar diversas canciones con el vídeo, se decide que la mejor opción es "*Happy Ukulele Whistle*" de AnduLeo. Con esto, se añade la canción al *timeline* y se ajusta de forma que se reproduzca varias veces (para cubrir todo el vídeo) sin que deje de sonar por completo y sea muy evidente el corte entre una reproducción y la siguiente (ver ilustraciones 48 y 49). Así pues, se ajusta el nivel

de volumen al principio y final de cada corte de la canción, para realizar un ajuste de dB y que se perciba un volumen uniforme a lo largo de todo el video. Este el único efecto aplicado al sonido, ya que se considera que la música original transmite lo que se pretendía desde un principio.

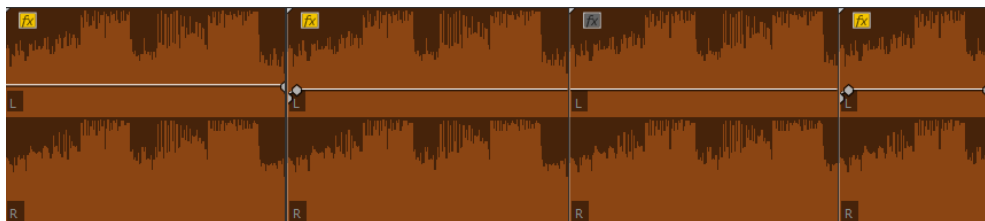


Ilustración 48 Edición de la música. Elaboración propia

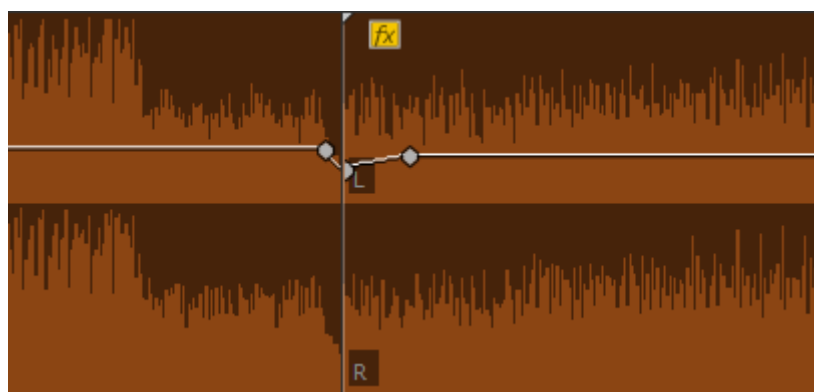


Ilustración 49 Ejemplo transición entre un clip de la canción y el siguiente. Elaboración propia

Este ajuste de la música se realiza en base al vídeo que engloba a las cuatro localizaciones. Sin embargo, posteriormente, la música será ajustada a los vídeos individuales para que en estos también se note un inicio y final de la canción y esta no corte bruscamente.

7.4 Corrección de color

La corrección del color es la parte más subjetiva del proceso de la edición, dado que no existen unos criterios de corrección establecidos, sino que dependen del autor. En este caso, teniendo en cuenta que el material se ha grabado en exterior y con luz natural, se decide que la corrección de color sea mínima, con tal de evitar llevar el color a un punto forzado y antinatural.

Por tanto, teniendo esto en cuenta, se crea una capa de ajuste por cada clip al que se iba a aplicar una corrección de color, con el objetivo de tener el *timeline* más ordenado y limpio. Además, a las capas de ajuste se les asigna una etiqueta de color verde, para que así sea más visual y fácil de identificar (ver ilustración 50).

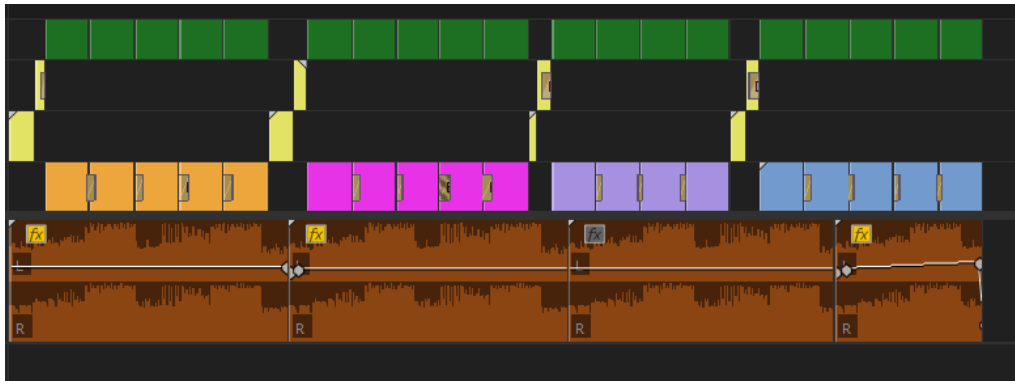


Ilustración 50 Timeline ordenado. Elaboración propia

Una vez ordenado el *timeline*, se varía un poco el valor de la exposición, reduciéndola al 85%, y manteniendo la misma en todos los clips. Además, se ajustan los valores de la exposición. La exposición es un valor que va de 0 a 100, siendo 0 el blanco absoluto y 100 el negro absoluto, siendo nula la cantidad de detalles en ambos extremos.

Para ajustar la exposición se toma como ayuda el sistema de zonas del fotógrafo Ansel Adams (1902-1984), mediante el cual se puede medir la escala de luz en once zonas distintas, en función del color y su pureza, siendo la zona 0 el negro puro y la zona X el blanco puro (ver ilustración 51).

Zona	Tono	Descripción
Zona 0	Negro puro	Negativo sin exponer, ninguna densidad útil
Zona I	Casi negro	Umbral efectivo. Primer paso antes del negro puro en la impresión, con ligera tonalidad, pero sin textura
Zona II	Negro oscuro	Primera insinuación de textura. Tonalidades profundas que representan la parte más oscura de la imagen en la cual se requiere poco detalle
Zona III	Gris muy oscuro	Materiales con oscuridad promedio y valores bajos que muestran textura adecuada
Zona IV	Gris poco oscuro	Vegetación con oscuridad promedio, rocas oscuras o sombras del paisaje. Valor normal de sombras para retratos de personas de tez blanca en luz intensa
Zona V	Gris intermedio	Gris con 18 % de reflectancia. Cielo claro boreal como se obtiene de película panorámica, tez oscura, roca gris, y madera oscurecida a la intemperie
Zona VI	Gris casi claro	Valor de la tez blanca en luz natural, luz natural o artificial difusa. Roca iluminada, sombras en la nieve en paisajes iluminados por el sol, claro cielo boreal en película panorámica con filtro azul claro
Zona VII	Gris claro	Tez muy blanca, objetos de color gris claro, nieve promedio con iluminación lateral adecuada
Zona VIII	Gris/Blanco pálido	Blancos con textura y valores delicados, nieve con textura, brillos en la tez blanca
Zona IX	Blanco brillante	Blanco sin textura aproximándose al blanco puro, con algo de tonalidad, pero sin textura. La nieve un día muy soleado.
Zona X	Blanco puro	Base papel fotográfico; brillo especular o fuentes de luz en la imagen

Ilustración 51 Sistema de zonas de Ansel Adams. Fuente: <https://claroscurocolombia.wordpress.com/portfolio/clase-7-sistema-de-zonas-modos-de-medicion-y-modos-de-enfoque-cargar-y-descargar-una-camara-analoga/>

Por tanto, observando la gráfica de la forma de onda de cada clip (ver ilustración 52) se asignan los valores de blancos y negros de cada clip siguiendo las recomendaciones de Ansel Adams.

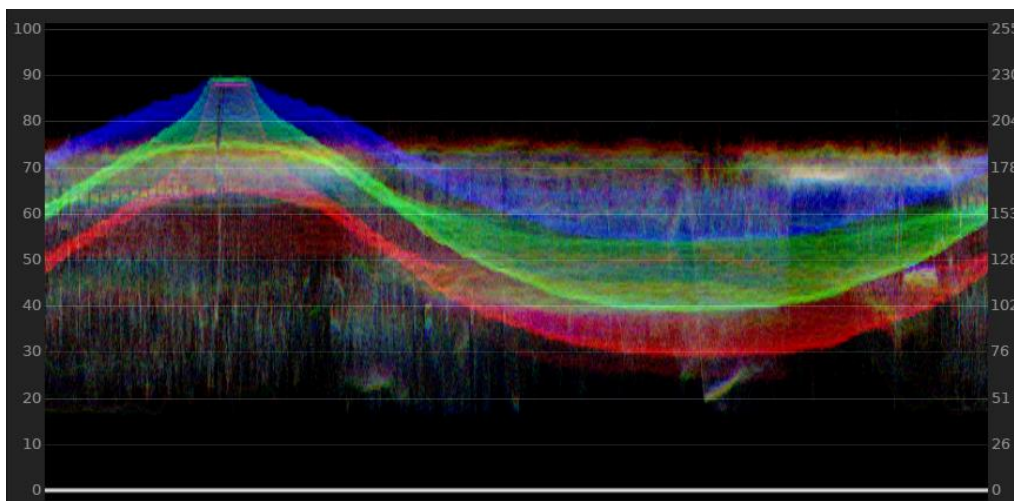


Ilustración 52 Ejemplo gráfica forma de onda. Elaboración propia

Tras este proceso de edición se obtiene el vídeo final, el cual es la base de trabajo de los siguientes apartados del proyecto.

7.5 Exportación

Tal y como se ha comentado anteriormente, se quieren obtener cinco vídeos, uno por cada una de las cuatro localizaciones y, posteriormente, un vídeo que englobe las cuatro localizaciones.

A la hora de la exportación del trabajo, se exporta el vídeo completo exportando todo el contenido del *timeline*. Además, para exportar los vídeos individuales de cada localización, se indican los instantes de inicio y fin sobre los fragmentos que se quiere exportar. Cabe remarcar que, como se ha comentado anteriormente, para los videos individuales se ajusta la música para cada una de las duraciones y espacios requeridos.

Sobre los ajustes de exportación de los vídeos, el aspecto más relevante y destacable es que se exportan en un formato H.264, ya que el códec utilizado define la tasa de compresión del vídeo y se considera que para videos en 8K el H.264 es la mejor opción. El códec superior sería el H.265 pero este da problemas de exportación y no

permite reproducir los vídeos en 360°. Con todo esto, se tiene los ajustes de exportación utilizados son los mostrados en la ilustración 53.

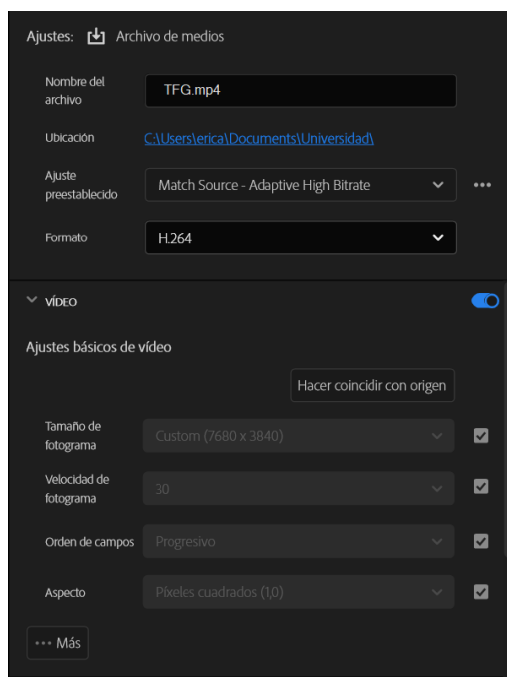


Ilustración 53 Ajustes de exportación. Elaboración propia

Teniendo ya los vídeos exportados se da por finalizado el proceso de grabación y posproducción de estos, dejando el material listo para seguir con la siguiente fase del proyecto.

8. Proceso de codificación

El objetivo de este apartado es preparar el contenido 360° generado para utilizar *streaming* adaptativo basado en HTTP (HAS) y la técnica *tile-based streaming*. Lamentablemente, por problemas con la herramienta utilizada no se ha podido aplicar ambas técnicas de forma simultánea. Sin embargo, se adjunta un anexo con una demo del uso de la técnica HAS, y a través del apartado se explican los pasos a seguir para aplicar ambas técnicas, además de mostrar el uso de la técnica *tile-based streaming*.

8.1 Generación de los vídeos en distintas calidades

Una vez editado el vídeo el siguiente paso es codificar dicho vídeo y hacerlo compatible con la tecnología DASH (Dynamic Adaptive *Streaming* over HTTP).

Antes de dividir el video en *tiles*, se debe codificar el vídeo en distintas calidades. Este proceso se lleva a cabo a través de la herramienta `ffmpeg`. En este ejemplo el vídeo se codifica en tres calidades distintas, con el *bitrate* y la resolución que se indican en la

tabla 8. En el anexo II se muestra el ejemplo de una demo donde se codifican más calidades.

Calidad	Resolución	Bitrate (Mbps)
1080p	1920x1080	8
2k	2048x1080	16
4k	3480x2160	35

Tabla 8 Relación resoluciones y bitrate en función de la calidad. Elaboración propia. Fuente: <https://support.google.com/youtube/answer/1722171?hl=es-419#zippy=%2Ctasa-de-bits>

Así pues, por cada calidad se introduce el siguiente comando, que son explicados en la tabla 9:

```
ffmpeg -i TFG.mp4 -r 30 -g 90 -keyint_min 90 -codec:v libx264 -profile:v baseline -level 13 -b:v 8000k -s 1920x1080 -vf drawtext="fontsize=80:fontfile=fuente.ttf:text='1080p':x=50:y=50:box=1:boxcolor=black @0.2" TFGoutput1080.mp4
```

Parámetro	Concepto
-i(input)	Se indica el fichero original
-r(rate) 30	Frames por segundo. 30 fps es el valor del video original, el cual se mantiene
-g y -keyint_min 90	Se indica fija el intervalo entre los llamados <i>keyframes</i> . Es importante ajustarlo ya que luego se dividirá el vídeo en segmentos, y cada segmento debe empezar por un <i>keyframe</i> (3 segundos por 30 frames/segundo = 90 frames.)
-codec:v libx264 - profile:v baseline -level 13	Se utilizará la biblioteca x264 para realizar la codificación con base lineal
-b(itrate):v	Se ajusta el valor del <i>bitrate</i> en función de los valores de la tabla 8
-s	Redimensiona el video en función de los valores de la tabla 8

-v(ideo)f(ilter)	Filtro de vídeo para escribir en la esquina superior izquierda el nombre de cada representación, con una fuente que se proporciona, tamaño 80, en las posiciones x = 50 e y = 50 y con una caja negra y transparente de fondo.
TFGoutput1080.mp4	Nombre del archivo de salida

Tabla 9 Explicación de cada uno de los parámetros. Elaboración propia

A partir de este parámetro se varía los atributos resaltados en negrita en función de la calidad a codificar, quedando de la siguiente forma los comandos para las otras dos calidades:

```
ffmpeg -i TFG.mp4 -r 30 -g 90 -keyint_min 90 -codec:v libx264 -profile:v baseline -level
13 -b:v 16000k -s 2048x1080 -vf
drawtext="fontsize=80:fontfile=fuente.ttf:text='2k':x=50:y=50:box=1:boxcolor=black@0.
2" TFGoutput2.mp4
```

```
ffmpeg -i TFG.mp4 -r 30 -g 90 -keyint_min 90 -codec:v libx264 -profile:v baseline -level
13 -b:v 35000k -s 3840x2160 -vf
drawtext="fontsize=80:fontfile=fuente.ttf:text='4k':x=50:y=50:box=1:boxcolor=black@0.
2" TFGoutput4.mp4
```

8.2 División en *tiles*

Una vez se tiene el vídeo codificado en las tres calidades distintas, también mediante la herramienta ffmpeg se cambia el formato del video de *mp4* a *yuv*, ya que el codificador que se va a utilizar para dividir el video en *tiles*, requiere que el vídeo de entrada sea *.yuv*. Para ello se introducen los siguientes comandos

```
ffmpeg -i TFGoutput1080.mp4 TFG1080.yuv
ffmpeg -i TFGoutput2.mp4 TFG2.yuv
ffmpeg -i TFGoutput4.mp4 TFG4.yuv
```

Con esto ya se tienen los videos de las tres calidades distintas listos para dividirlo en *tiles* y hacerlo compatible con la tecnología DASH. Este proceso se ha llevado a cabo a

través del codificador de código abierto Kvazaar. Para trabajar con este codificador se tienen dos opciones, o trabajar directamente desde Linux o trabajar a través de WSL (Windows Subsystem for Linux).

8.2.1 Activación del sistema WSL (*Windows Subsystem for Linux*)

En este caso, al trabajar desde un equipo con Windows como sistema operativo se toma la segunda opción y, lo primero que se hace es instalar Ubuntu en el dispositivo desde la *Microsoft Store* y, a continuación, se debe activar WSL, a través de “Windows PowerShell (Administrador)” haciendo clic derecho sobre el botón Inicio.

Esto hace que se abra la ventana mostrada en la Ilustración 54, en la cual se debe introducir el siguiente comando y, posteriormente, reiniciar el ordenador pulsando la tecla Intro. Una vez reiniciado el ordenador, al abrir el *software* Ubuntu se inicia el proceso de instalación, durante el cual se pide un usuario y una contraseña.

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
```

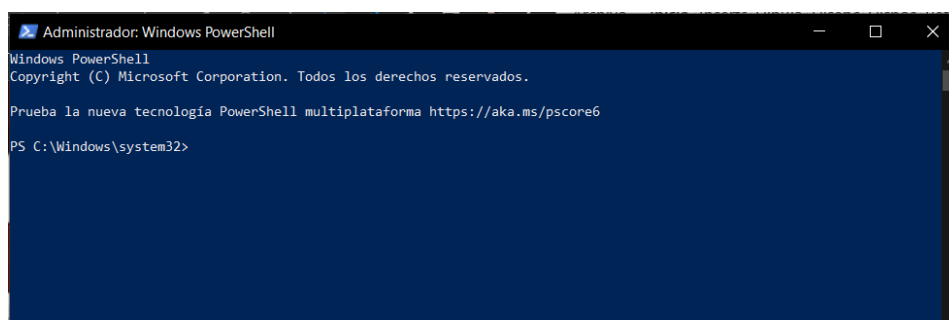


Ilustración 54 Consola PowerShell de Windows. Elaboración propia

8.2.2 Instalación del codificador Kvazaar

Con esto ya se tiene todo listo para poder codificar el vídeo. En este caso, como ya se ha comentado anteriormente, la codificación se realiza con el codificador de código abierto Kvazaar. Para trabajar con este codificador se debe clonar, desde Windows, el directorio de GitHub a través de la propia aplicación de escritorio de GitHub para así tener el proyecto almacenado en el dispositivo. Para clonar el directorio se tiene que instalar el *software* GitHub desktop para Windows y desde la opción File acceder a “Clone repository” (ver Ilustración 55).

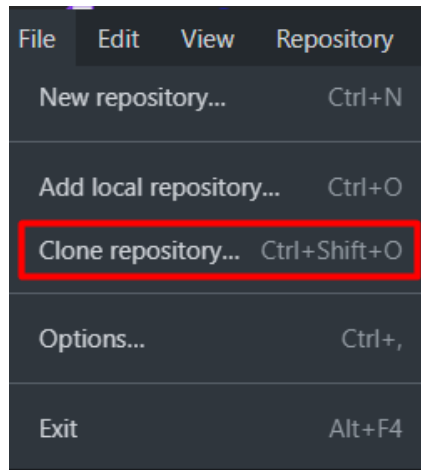


Ilustración 55 Instrucciones para clonar repositorio de GitHub. Elaboración propia

Finalmente, desde la pestaña URL se añade la URL del directorio a clonar (en este caso <https://github.com/ultravideo/kvazaar>) y se elige donde se va a guardar el proyecto (ver Ilustración 56).

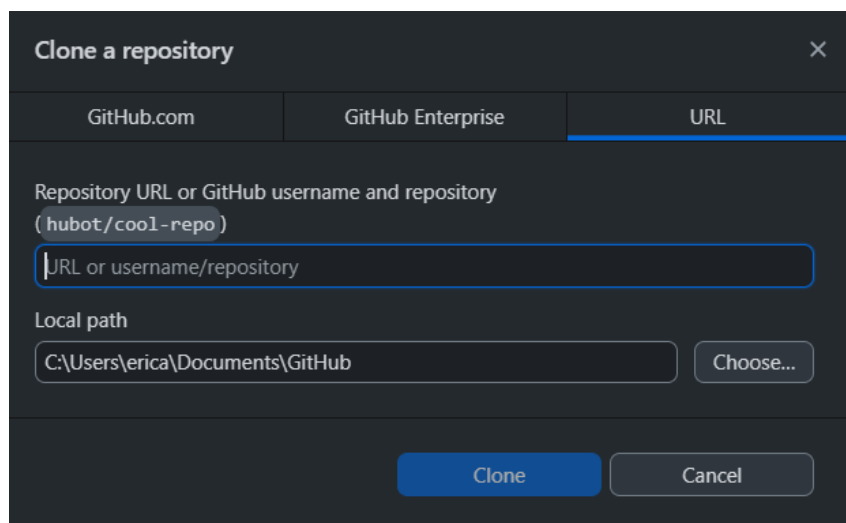


Ilustración 56 Menú de opciones para clonar el repositorio desde la app Github para Windows. Elaboración propia

A continuación, se debe compilar e instalar Kvazaar, para ello se siguen los siguientes pasos:

1. Instalar las herramientas autotools, automake, autoconf, libtool, m4 y yasm. Para ello se utiliza el comando `sudo apt-get install herramienta`, siendo herramienta la que se vaya a instalar en ese momento y, repitiendo el proceso para cada una de ellas.

2. Instalar un compilador de C, a través del comando *sudo apt-get install build-essential*,

3. Únicamente si se está trabajando a través de WSL se deben introducir los siguientes comandos para poder realizar el paso 5 sin errores.

```
$ sudo apt-get install dos2unix
$ dos2unix *.am *.ac *.sh m4/*
$ dos2unix src/*.h src/*.c
```

4. Cambiar el directorio a la carpeta donde se encuentra el archivo “makefile” para poder compilar en C. En este caso el directorio sobre el que se trabaja es */mnt/c/Users/erica/Documents/GitHub/kvazaar*.

Al trabajar sobre Ubuntu se debe adaptar la ruta de Windows (C:\Users\erica\Documents\GitHub\kvazaar) al formato leído por Ubuntu (añadiendo el */mnt/* al principio, poniendo la ‘c’ en minúscula y sin los dos puntos, e invirtiendo las barras).

5. Introducir los siguientes comandos para compilar e instalar Kvazaar:

```
./autogen.sh
./configure
make
sudo make install
sudo ldconfig
```

8.2.3 Creación de *tiles*

Realizando estos pasos ya se ha compilado e instalado Kvazaar, por lo que se procede a dividir cada uno de los vídeos en *tiles*. En este caso dichos *tiles* tienen un tamaño de 3x3. Con esto se tiene que los tres comandos, uno para cada vídeo de distinta calidad, son los siguientes:

```
kvazaar -i TFG1080.yuv --input-res 1920x1080 -o TFG1080.hvc --tiles 3x3 --slices tiles
--mv-constraint frametilemargin --bitrate 8000000 --period 30 --input-fps 30
```

kvazaar -i TFG2.yuv --input-res 2048x1080 -o TFG2.hvc --tiles 3x3 --slices tiles --mv-constraint frametilemargin --bitrate 16000000 --period 30 --input-fps 30

kvazaar -i TFG4.yuv --input-res 3840x2160 -o TFG4.hvc --tiles 3x3 --slices tiles --mv-constraint frametilemargin --bitrate 35000000 --period 30 --input-fps 30

En la tabla 10, se explican los parámetros utilizados.

Parámetro	Concepto
<i>-i(input)</i>	Se indica el fichero original
<i>--input-res</i>	Se indica la resolución del fichero original
<i>-o</i>	Nombre del archivo de salida
<i>-tiles 3x3 --slices tiles</i>	Se indica el tamaño de la división de tiles
<i>-mv-constraint frametilemargin</i>	Indica que los vectores de movimiento son restringidos dentro de cada <i>tile</i>
<i>-bitrate</i>	Se indica el bitrate del video en bps, acorde a los valores de la tabla 8
<i>-input-fps 30</i>	Se indica el valor de frames por segundo del video original

Tabla 10 Explicación parámetros del proceso de creación de tiles. Elaboración propia

8.3 Generación de segmentos DASH y su mpd

8.3.1 Empaquetar el flujo de bits

El siguiente paso consiste en empaquetar el flujo de bits HEVC a través de la herramienta MP4Box.

MP4Box -add TFG1080.hvc:split_tiles -fps 30 -new TFG1080_tiled.mp4

MP4Box -add TFG2.hvc:split_tiles -fps 30 -new TFG2_tiled.mp4

MP4Box -add TFG4.hvc:split_tiles -fps 30 -new TFG4_tiled.mp4

8.3.2 Generación segmentos DASH y su mpd

Finalmente, también a través de MP4Box, se generan los segmentos de DASH y el archivo .mpd. En la tabla 11, se explican los parámetros utilizados.

MP4Box -dash 3000 -rap -frag-rap -profile live -out 1080dash_tiled.mpd
 TFG1080_tiled.mp4
 MP4Box -dash 3000 -rap -frag-rap -profile live -out 2dash_tiled.mpd TFG2_tiled.mp4
 MP4Box -dash 3000 -rap -frag-rap -profile live -out 4dash_tiled.mpd TFG4_tiled.mp4

Parámetro	Concepto
-dash 3000	Indica que se está segmentando el vídeo para utilizar DASH en segmentos de 3000 ms (3s)
-rap	Indica que cada segmento comenzará obligatoriamente con un RAP (<i>Random Access Point</i>), que permitirá acceder a cada segmento al desplazar la reproducción o cambiar de calidad.
-frag-rap	Crea subsegmentos de los segmentos
-profile live	Se indica la plantilla que se va a usar para generar los segmentos
-out	Indica el nombre del archivo de salida
TFG4_tiled.mp4	Se indica el nombre del archivo de origen

Tabla 11 Explicación parámetros generación del archivo .mpd. Elaboración propia

Para obtener el archivo .mpd de las tres calidades juntas se debe introducir el siguiente comando:

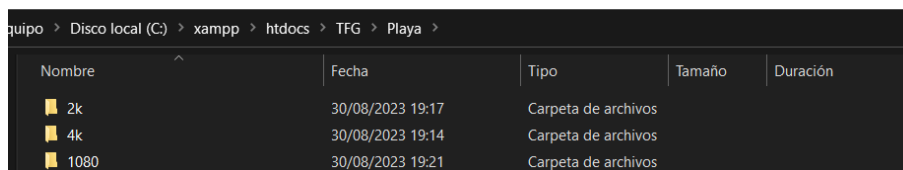
MP4Box -dash 1000 -rap -frag-rap -profile live -out dash_tiled.mpd TFG080_tiled.mp4
 TFG2_tiled.mp4 TFG4_tiled.mp4

No obstante, tal y como se ha comentado al inicio del apartado, la reproducción a partir de este archivo .mpd no se ha podido conseguir por problemas de la herramienta con la reproducción de archivos .mpd que incluyan múltiples representaciones.

8.4 Creación del servidor HTTP y almacenamiento de los segmentos

Conforme se van generando los segmentos se ordenan por carpetas en función de cada una de las calidades (ver ilustración 57). Dichas carpetas con las calidades se deben

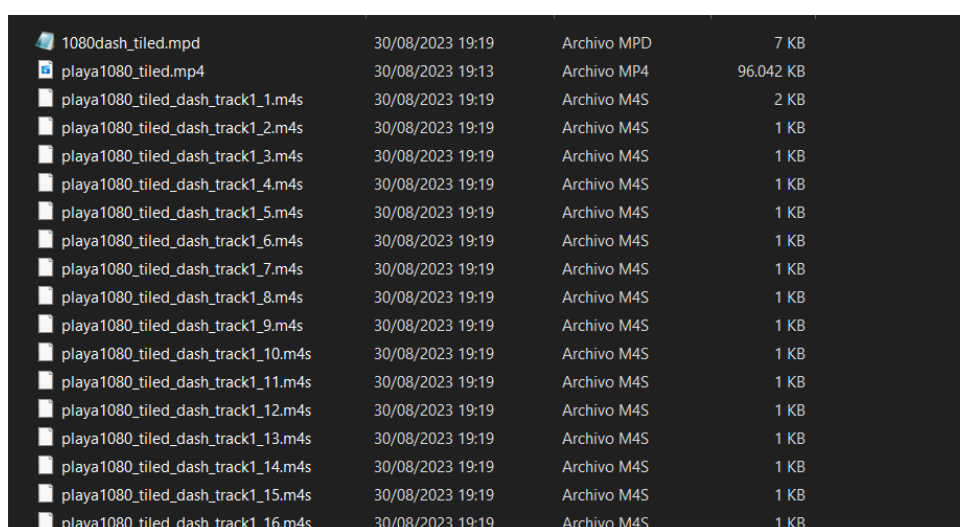
almacenar en un servidor HTTP. Para ello, se crea un servidor con la herramienta XAMPP⁵ y se almacenan las carpetas de las distintas calidades en la carpeta “htdocs/TFG/Localización” del propio servidor.



Nombre	Fecha	Tipo	Tamaño	Duración
2k	30/08/2023 19:17	Carpeta de archivos		
4k	30/08/2023 19:14	Carpeta de archivos		
1080	30/08/2023 19:21	Carpeta de archivos		

Ilustración 57 Organización de los segmentos en función de la calidad. Ejemplo localización playa. Elaboración propia

Dentro de cada carpeta se almacenan los distintos segmentos de cada calidad junto a su archivo .mpd correspondiente (ver ilustración 58).



Nombre	Fecha	Tipo	Tamaño
1080dash_tiled.mpd	30/08/2023 19:19	Archivo MPD	7 KB
playa1080_tiled.mp4	30/08/2023 19:13	Archivo MP4	96.042 KB
playa1080_tiled_dash_track1_1.m4s	30/08/2023 19:19	Archivo M4S	2 KB
playa1080_tiled_dash_track1_2.m4s	30/08/2023 19:19	Archivo M4S	1 KB
playa1080_tiled_dash_track1_3.m4s	30/08/2023 19:19	Archivo M4S	1 KB
playa1080_tiled_dash_track1_4.m4s	30/08/2023 19:19	Archivo M4S	1 KB
playa1080_tiled_dash_track1_5.m4s	30/08/2023 19:19	Archivo M4S	1 KB
playa1080_tiled_dash_track1_6.m4s	30/08/2023 19:19	Archivo M4S	1 KB
playa1080_tiled_dash_track1_7.m4s	30/08/2023 19:19	Archivo M4S	1 KB
playa1080_tiled_dash_track1_8.m4s	30/08/2023 19:19	Archivo M4S	1 KB
playa1080_tiled_dash_track1_9.m4s	30/08/2023 19:19	Archivo M4S	1 KB
playa1080_tiled_dash_track1_10.m4s	30/08/2023 19:19	Archivo M4S	1 KB
playa1080_tiled_dash_track1_11.m4s	30/08/2023 19:19	Archivo M4S	1 KB
playa1080_tiled_dash_track1_12.m4s	30/08/2023 19:19	Archivo M4S	1 KB
playa1080_tiled_dash_track1_13.m4s	30/08/2023 19:19	Archivo M4S	1 KB
playa1080_tiled_dash_track1_14.m4s	30/08/2023 19:19	Archivo M4S	1 KB
playa1080_tiled_dash_track1_15.m4s	30/08/2023 19:19	Archivo M4S	1 KB
playa1080_tiled_dash_track1_16.m4s	30/08/2023 19:19	Archivo M4S	1 KB

Ilustración 58 Ejemplo segmentos almacenados en la subcarpeta 1080

8.5 Reproducción del contenido

Para poder reproducir los contenidos a partir del archivo índice generado (manifest.mpd) mediante el sistema WSL primero se debe actualizar el Ubuntu 22.04 a la versión 2. Para ello en la consola de PowerShell se introduce `wsl --set-version Ubuntu-22.04 2` y, posteriormente, con el comando `wsl --list --verbose` se comprueba que se ha actualizado a la versión 2 (ver Ilustración 59).

```
PS C:\Users\erica> wsl --list --verbose
NAME                STATE              VERSION
* Ubuntu-22.04      Running            2
```

Ilustración 59 Menú de opciones para clonar el repositorio. Elaboración propia

⁵ <https://www.apachefriends.org/es/download.html>

Finalmente, en la consola de Ubuntu se debe introducir el siguiente comando:

```
gpac -gui URL#VR
```

Por ejemplo, para reproducir el video de la localización de la playa a una calidad de 2k, el comando queda de la siguiente forma:

```
gpac -gui Dir_IP_del_servidor_web/TFG/Playa/2k/2dash_tiled.mpd#VR
```

Al reproducir el contenido, se debería observar que la imagen se divide en nueve *tiles* (al indicar una división 3x3) donde los tres *tiles* del centro se reproducen a mayor calidad que los seis *tiles* de los extremos, adaptándose conforme se gira la imagen. Se lamenta que no se haya podido reproducir el archivo .mpd que contiene las tres calidades distintas y no se pueda observar la división de tiles. No obstante, se adjunta un anexo con una demo del uso de la técnicas HAS, teniendo así la representación de ambas técnicas en el proyecto.

9. Inclusión de efectos multisensoriales

9.1 Instalación *software* SEVino2

Para la instalación del programa SEVino2 se van a requerir diversos componentes que se indicarán a continuación. Pero, en primer lugar, se debe descargar el propio *software* SEVino2 desde la siguiente página web: <https://sourceforge.net/projects/sevino/>

Dado que este *software* de generación de contenidos mulsemmedia se encuentra desactualizado desde el año 2014, se requieren una versión de Java antigua para poder ejecutar el programa correctamente. Por tanto, el siguiente paso es descargar la versión jdk1.80_74, concretamente la versión de 64 bits (ver ilustración 60), descargable desde el siguiente enlace: <https://www.oracle.com/java/technologies/javase/javase8-archive-downloads.html>

Solaris x64 (SVR4 package)	140.02 MB	jdk-Bu74-solaris-x64.tar.Z
Solaris x64	96.19 MB	jdk-Bu74-solaris-x64.tar.gz
Windows x86	182.01 MB	jdk-Bu74-windows-i586.exe
Windows x64	187.31 MB	jdk-Bu74-windows-x64.exe

Ilustración 60 Versión de Java a descargar. Elaboración propia

Para la automatización de compilación desarrollada en lenguaje Java, en la construcción de proyectos, es necesario descargar Apache Ant, en este caso la versión descargada ha sido la 1.10.13, disponible en la página web: <https://ant.apache.org/bindownload.cgi>

Finalmente, el último *software* a descargar es el reproductor de vídeo VLC, mediante el cual se reproducirá el contenido en el *software* SEVino2. Se debe tener en cuenta, que para que el programa se ejecute se debe descargar una versión de 64 bits, al igual que la versión de Java que se va a utilizar. En esta ocasión se ha descargado la versión 3.0.8, descargable desde el siguiente enlace: <http://get.videolan.org/vlc/3.0.8/win64/vlc-3.0.8-win64.exe>

Una vez descargado todo el *software* necesario, se procede a añadir las correspondientes variables del sistema. Para acceder a dichas variables se realizan los pasos de las ilustraciones 61 y 62.

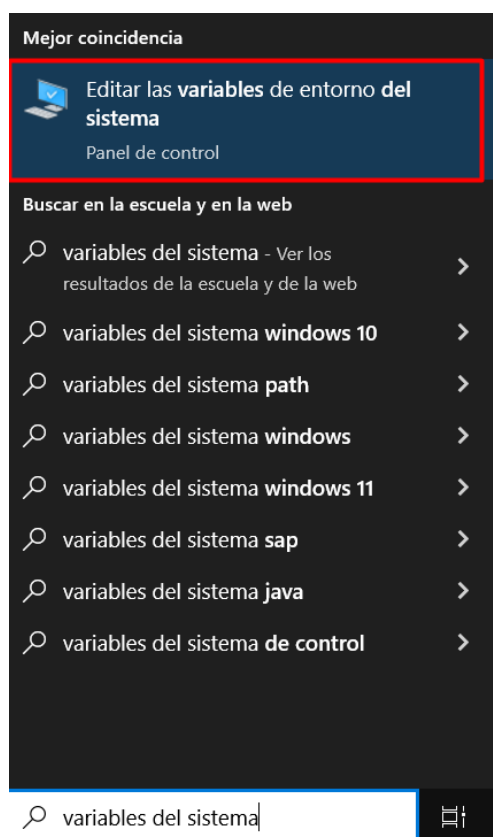


Ilustración 61 Acceder a las variables del sistema (Paso 1). Elaboración propia

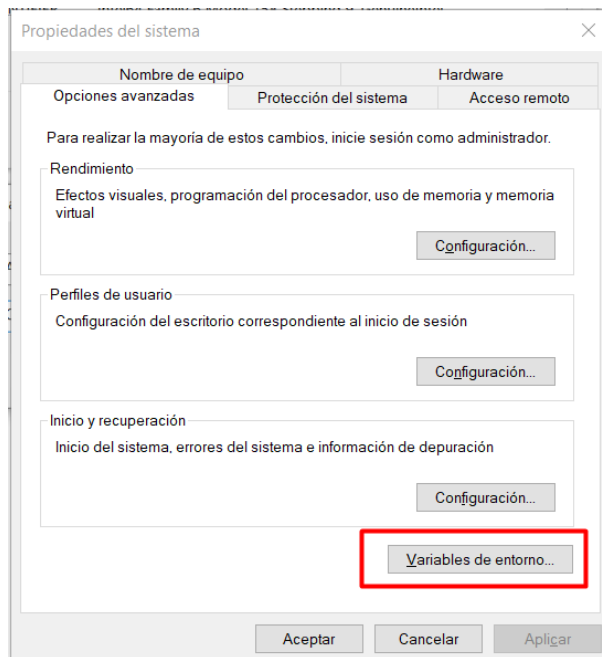


Ilustración 62 Acceder a las variables del sistema (Paso 2).
Elaboración propia

A continuación, con el botón “Nueva...” se deben añadir las variables ANT_HOME y JAVA_HOME, tal y como se indican en la ilustración 63.

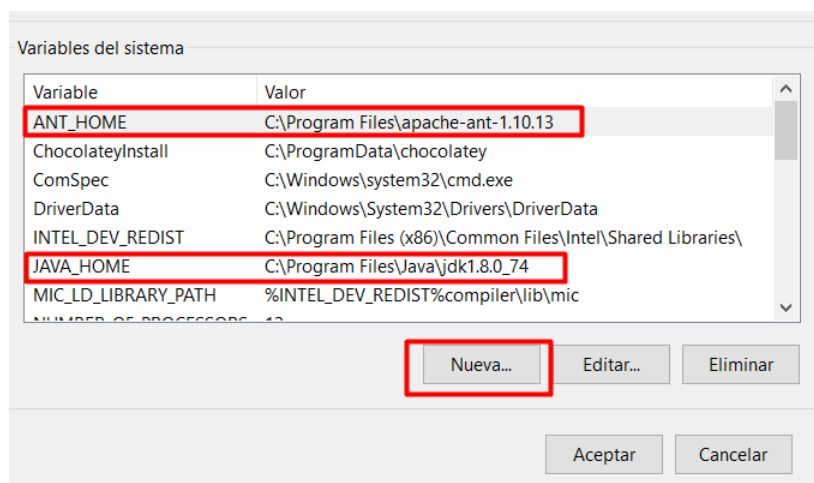


Ilustración 63 Ejemplo variables ANT_HOME y JAVA_HOME. Elaboración propia

Finalmente teniendo la variable “path” seleccionada se debe editar dicha variable y con el botón “Nuevo”, se debe añadir la ruta hasta los archivos previamente descargados, tal y como se indica en la ilustración 64.

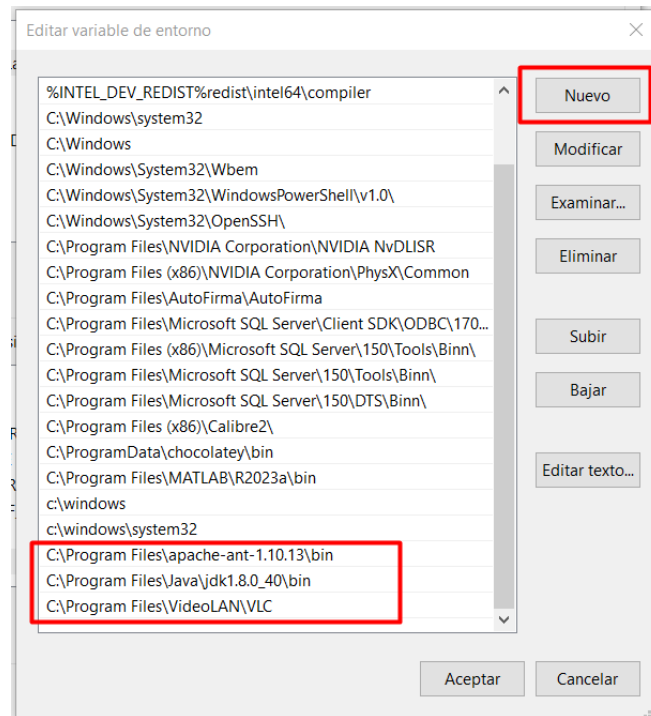


Ilustración 64 Ejemplo variables path. Elaboración propia

Finalmente, para terminar con el proceso de instalación, se debe crear un fichero de texto llamado SEVino2 en la misma carpeta donde se ha descargado dicho *software*. El archivo contendrá una ruta al directorio en el cual se encuentra el reproductor de vídeo VLC, y será el encargado de ejecutar el programa. Para ello, se debe introducir lo indicado en la ilustración 65 y, posteriormente, cambiar el archivo de un formato .txt a un formato .bat.

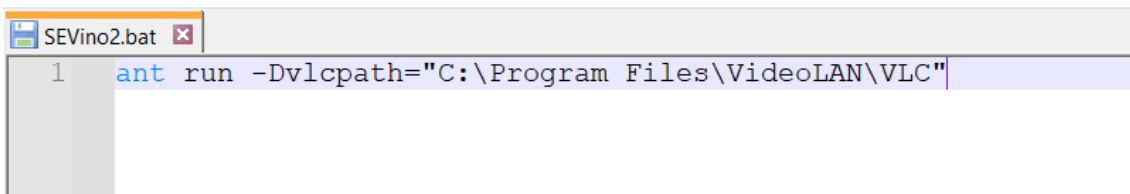


Ilustración 65 Ejemplo archivo SEVino2.bat. Elaboración propia

Con tal de ejecutar el programa SEVino2, se debe abrir la consola del sistema y se debe acceder a la ruta donde se encuentra el programa y, una vez en dicha ruta se debe ejecutar el comando "SEVino2.bat". Por ejemplo, para este caso se ejecutarían los comandos mostrados en la ilustración 66.

```
Microsoft Windows [Versión 10.0.19045.3086]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\erica>cd C:\Users\erica\Documents\SEVino2

C:\Users\erica\Documents\SEVino2>SEVino2.bat

C:\Users\erica\Documents\SEVino2>ant run -Dvlcpath="C:\Program Files\VideoLAN\VLC"
ANT_OPTS is set to -Djava.security.manager=allow
Buildfile: C:\Users\erica\Documents\SEVino2\build.xml
```

Ilustración 66 Iniciar software SEVino2. Elaboración propia

9.2 Creación de efectos multisensoriales

Al abrir la aplicación SEVino2 se muestra la ventana de la ilustración 67, donde a través de la opción open AV, se debe abrir el vídeo al que se le desee añadir los distintos efectos sensoriales.

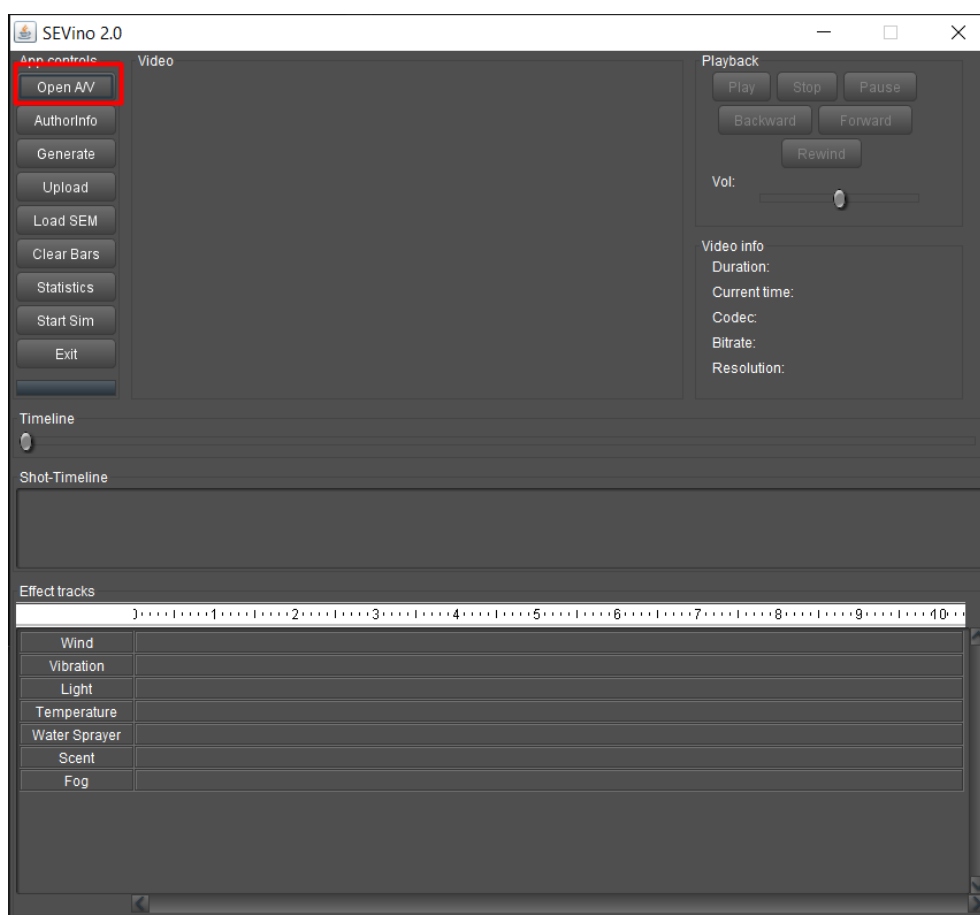


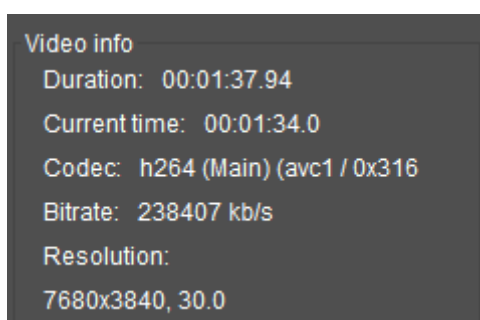
Ilustración 67 Ventana de inicio aplicación SEVino2. Elaboración propia

Al cargar un vídeo, en la parte superior derecha se activarán los controles de reproducción y la opción de ajustar el volumen del propio video (ver ilustración 68).



*Ilustración 68 Controles de reproducción y volumen.
Elaboración propia*

También en el lateral derecho, debajo de estos controles se muestra información del vídeo que se ha cargado. Concretamente se muestra la duración, el instante de tiempo que se está reproduciendo, el códec, el *bitrate* y la resolución del vídeo utilizado (ver ilustración 69).



*Ilustración 69 Información del video.
Elaboración propia*

Debajo de la pantalla de reproducción, se encuentra el *timeline* que permite retroceder y adelantar manualmente el vídeo. También, se encuentra el *shot-timeline* donde, normalmente, se muestra fragmentado el vídeo por escenas. Sin embargo, en esta ocasión, al tratarse de un video 360º no se encuentra fragmentado (ver ilustración 70).

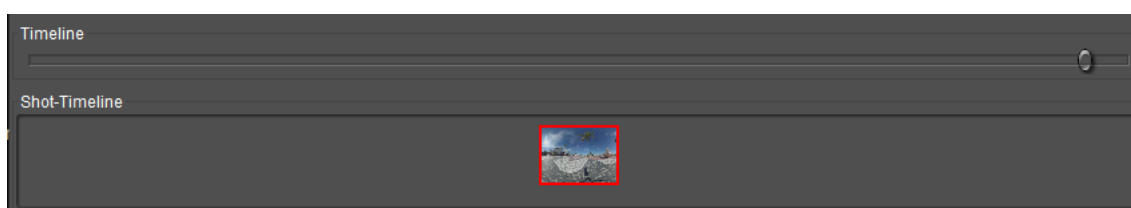


Ilustración 70 Timeline y shot-timeline. Elaboración propia

Finalmente, en la parte inferior de la aplicación se encuentra el Effect tracks, que es donde se agregarán los efectos sensoriales al vídeo (ver ilustración 71). Los efectos sensoriales que dispone el programa son: efecto de viento, de vibración, de luz, de temperatura, de humedad a través de un rociador de agua, de aroma y de niebla. En este proyecto el efecto utilizado es el aroma, contando con un difusor de aromas profesional para la inclusión de los distintos aromas.

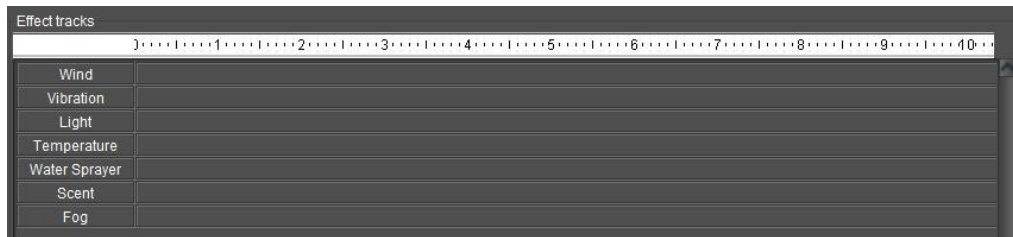


Ilustración 71 Pista de efectos. Elaboración propia

Para añadir un efecto sensorial se debe hacer clic en la pista de efectos y arrastrarlo a lo largo del efecto el cuál se desea agregar. Al realizar esta acción se muestra una ventana emergente (ver ilustración 72) en la que se pueden definir los siguientes parámetros que afectan al efecto sensorial:

- **Duration:** Indica la duración del efecto sensorial.
- **Alt:** Permite indicar un efecto alternativo por si el efecto original no se puede generar debido a la falta de dispositivos.
- **Priority:** Indica la prioridad de los efectos respecto a otros efectos que comparten el mismo instante de tiempo.
- **Location:** Describe la ubicación desde donde se espera que se reciba el efecto.
- **Intensity-value:** Mediante un valor comprendido en una escala indica la potencia del efecto que se va a generar.
- **Intensity-range:** Permite la adaptación con un límite inferior y superior determinados. Por ejemplo, del 0 al 100.
- **AutoExtraction:** Se utiliza para indicar si se extrae de forma automática el efecto sensorial del recurso multimedia.
- **Start time:** Indica el comienzo del efecto sensorial.
- **End time:** Indica el fin del efecto sensorial.

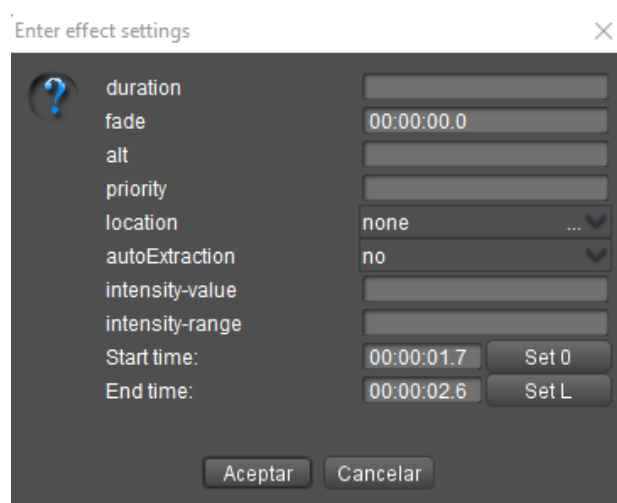


Ilustración 72 Ajustes de efectos. Elaboración propia

Una vez añadidos los efectos, se debe pulsar el botón *Generate* para generar el archivo SEM con los datos de los efectos sensoriales. El archivo SEM es un archivo de tipo .xml el cual se puede abrir para comprobar los efectos y los metadatos generados con SEVino2 y, también, comprobar que estos siguen la estructura del estándar MPEG-V (ver ilustración 73).

```
<?xml version="1.0" encoding="UTF-8" ?>
<ns2:SEM xmlns:ns2="urn:mpeg:mpeg-v:2010:01-SEDI-NS" xmlns:ns1="urn:mpeg:mpeg7:schema:2004" xmlns:ns3="urn:mpeg:mpeg-v:2010:01-SEV-NS" xmlns:ns4="urn:mpeg:mpeg-v:2010:01-SEV-NS" intensity-value="7.0" intensity-range="0.0 10.0" activate="true" si:anchorElement="true" si:pts="5481000"/>
<ns2:Effect xsi:type="ns3:ScentType" activate="false" si:anchorElement="true" si:pts="6858000"/>
<ns2:Effect xsi:type="ns3:ScentType" intensity-value="7.0" intensity-range="0.0 10.0" activate="true" si:anchorElement="true" si:pts="6957000"/>
<ns2:Effect xsi:type="ns3:ScentType" activate="false" si:anchorElement="true" si:pts="8802000"/></ns2:SEM>
```

Ilustración 73 Ejemplo archivo SEM generado con SEVino2. Elaboración propia

Con el archivo SEM generado el último paso es proporcionar dicho archivo al reproductor de vídeo 360°, el cual se encarga de enviar al difusor los mensajes UDP pertinentes para la reproducción de aromas. En el anexo II se adjunta un ejemplo detallado de como este proceso se lleva a cabo.

10. Conclusiones

A lo largo del proyecto, se ha aprendido tanto a manejar una cámara 360° como a realizar de manera correcta y coherente el proceso de edición y exportación del material grabado. Específicamente, a través del vídeo que incluye las cuatro localizaciones, se ha trabajado una forma de unir diversas localizaciones para obtener un vínculo y una narrativa gracias a la edición. Para ello, se han aprendido los conocimientos básicos del *software* Adobe After Effects, el cual nunca había sido utilizado anteriormente a este proyecto, por lo que no se tenía un conocimiento de este. Además, se ha profundizado en los conocimientos que se tenían previamente sobre el *streaming* adaptativo basado en HTTP y la técnica *tile-based streaming*. Se han aplicado estos conocimientos a lo largo de todo el trabajo para obtener el producto final, cumpliendo así con los objetivos establecidos al inicio del proyecto.

Como medidas de mejora para un futuro cercano, se tiene el hecho de optimizar la división en *tiles*, pues para la realización de este proyecto se ha trabajado con una división de 3x3, la cual quizás no es la división óptima para trabajar con videos 360°. Tal y como se ha comentado a lo largo del documento, el uso de la técnica *tile-coding* en videos 360° aún tiene un gran camino de pruebas e investigaciones con tal de descubrir cual sería la división en *tiles* perfecta y la optimización máxima.

Sin duda, el campo de los videos 360° seguirá creciendo en los próximos años y, con él, también lo harán las distintas técnicas y tecnologías que buscan la mejora constante para el usuario y su experiencia. Asimismo, a raíz del aumento del interés por parte de los usuarios de dejar de “ver” vídeos como tal para empezar a “sentir” la acción de estos, se espera que la inclusión de efectos multisensoriales y las tecnologías mulsemmedia también vayan en aumento pues tienen como objetivo principal el dotar al usuario de una experiencia más enriquecedora, tal y como se ha comentado en diversas ocasiones a lo largo del documento.

Como conclusión general se tiene que, pese a que no se han cumplido todos los objetivos establecidos al principio del proyecto por problemas con la herramienta GPAC para reproducir multi representaciones, se han adquirido una gran cantidad de conocimientos en la realización de este TFG. Así pues, estos objetivos no cumplidos son puntos abiertos en los que se puede profundizar la investigación y el trabajo realizado y así seguir mejorando el proyecto.

Bibliografía

- Adobe. (s.f.). Obtenido de <https://business.adobe.com/es/products/primetime/adobe-media-server/hds-dynamic-streaming.html>
- Apple, I. (Agosto de 2017). *RFC editor*. Obtenido de <https://datatracker.ietf.org/doc/html/rfc8216>
- Chiariotti, F. (2021). A survey on 360° video: coding, quality of experience and streaming. *Elsevier Computer Communications*.
- Cloudflare. (s.f.). Recuperado el 13 de Julio de 2023, de <https://www.cloudflare.com/es-es/learning/video/what-is-http-dynamic-streaming/>
- Cook, A. (2017). *Github*. Obtenido de https://github.com/udacity/deep-reinforcement-learning/blob/master/tile-coding/Tile_Coding_Solution.ipynb
- Estevao B. Saleme, A. C. (2019). Mulsemedia DIY: A survey of devices and a tutorial for Building your own mulsemedia environment. *ACM Computing Surveys*, 52(3).
- Grau, O. (2002). *Virtual Art: From Illusion to Immersion*. Massachusetts: The MIT Press.
- Guridi, P. F. (16 de Noviembre de 2022). *Video Adaptativo sobre HTTP*. Recuperado el 10 de Julio de 2023, de https://eva.fing.edu.uy/pluginfile.php/262934/mod_resource/content/4/slides-modA-has-20221116.pdf
- Hernández, P. (04 de Agosto de 2022). Obtenido de <https://filmora.wondershare.es/virtual-reality/difference-between-360-video-vr.html>
- Hernández, P. (04 de Agosto de 2022). *Filmora*. Recuperado el 2023 de Julio de 2, de <https://filmora.wondershare.es/virtual-reality/difference-between-360-video-vr.html>
- Jlasso. (10 de Noviembre de 2014). *Un secreto gigante*. Obtenido de <https://unsecretogigantesco.com/adaptive-streaming-eso-que-es/>
- Mai, H. (22 de Julio de 2019). Obtenido de <https://medium.com/criteo-engineering/tile-coding-an-efficient-sparse-coding-method-for-real-valued-data-e787eddf630a>
- Marketing audiovisual*. (21 de Octubre de 2022). Obtenido de <https://marketingaudiovisual.com/vertical/>
- Microsoft*. (2023 de Julio de 19). Obtenido de <https://learn.microsoft.com/es-es/iis/media/smooth-streaming/smooth-streaming-primer>
- Microsoft*. (s.f.). *Microsoft*. Obtenido de <https://www.microsoft.com/silverlight/>
- MPEG. (2014). *MPEG-DASH*. Obtenido de <https://www.mpeg.org/standards/MPEG-DASH/>
- MPEG-V Sensory Information*. (2009). (MPEG) Recuperado el 16 de Julio de 2023, de <https://mpeg.chiariglione.org/standards/mpeg-v/sensory-information>
- Mueller, C. (21 de Mayo de 2015). *Bitmovin*. Recuperado el 13 de Julio de 2023, de <https://bitmovin.com/microsoft-smooth-streaming-mss/>
- Mueller, C. (21 de Mayo de 2015). *Bitmovin*. Obtenido de <https://bitmovin.com/microsoft-smooth-streaming-mss/>
- Mueller, C. (21 de Mayo de 2015). *Bitmovin*. Obtenido de <https://bitmovin.com/microsoft-smooth-streaming-mss/>
- Nielsen, M. F., & Laura, d. A. (2022). *360: Diseño y Realización de Video Inmersivo*. Nobuko.
- Normas APA. (8 de Febrero de 2019). Obtenido de <https://normas-apa.org/wp-content/uploads/Guia-Normas-APA-7ma-edicion.pdf>
- Rai, D. (23 de Enero de 2020). *Youtube*. Recuperado el 13 de Julio de 2023, de https://www.youtube.com/watch?v=sy2Lbc_DYo4

- Rodriguez, P. (2 de Enero de 2019). *Xatakahome*. Recuperado el 2023 de Julio de 16, de <https://www.xatakahome.com/ocio/feelreal-vr-mask-quiery-extender-realidad-virtual-convencional-a-dimension-olfativa-tactil>
- Rytewiki. (2021). Obtenido de https://es.ryte.com/wiki/Video_360%C2%BA
- Saleme, E., Covaci, A., Mesfin, G., Santos, C., & Ghinea, C. (2019). Mulsemedia DIY: A survey of devices and a tutorial for building your own mulsemedia environment. *ACM Computing Survey*.
- Sangameshwarkar, N. C. (2017). Study of video streaming standards. *International Journal of Latest Engineering Research and Applications (IJLERA)*, II(6), 71-82.
- Sangameshwarkar, N. C. (2017). Study of video streaming standards. *International journal of latest engineering research and applications*, 71-82.
- Stack overflow. (9 de Mayo de 2016). Obtenido de <https://stackoverflow.com/questions/37023460/function-approximation-how-is-tile-coding-different-from-highly-discretized-sta>
- Telefónica. (s.f.). Recuperado el 13 de Julio de 2023, de <https://www.telefonicaserviciosaudiovisuales.com/articulos-de-divulgacion/protocolos-de-streaming/>
- Timmerer, C., Gelissen, J., Walti, M., & Hellwagner, H. (s.f.). *Interfacing with virtual worlds*. Recuperado el 16 de Julio de 2023, de <https://www.mpeg.org/wp-content/uploads/whitepapers/files/MPEG-V/MPEG-V.pdf>
- tour360. (4 de Agosto de 2022). *Tour360*. Obtenido de <https://www.tour360.pe/los-beneficios-del-contenido-360-en-el-sector-turismo/>
- Wikipedia. (s.f.). *Wikipedia*. Obtenido de https://es.wikipedia.org/wiki/%C3%81ngulo_s%C3%B3lido
- Yaqooh, A., Bi, T., & Muntean, G.-M. (2020). A survey on adaptive 360° video streaming: solutions, challenges and opportunities. *IEEE*.
- Zhang, J. (27 de Julio de 2019). Obtenido de <https://towardsdatascience.com/reinforcement-learning-tile-coding-implementation-7974b600762b>