

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA POLITECNICA SUPERIOR DE GANDIA

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



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCOLA POLITÈCNICA
SUPERIOR DE GANDIA

“Plataforma web multimedia para docencia interactiva y colaborativa a distancia”

TRABAJO FINAL DE GRADO

Autor/a:
Pau Salvador Llàcer

Tutor/a:
Fernando Boronat Seguí

GANDIA, 2018

ABSTRACT

Currently, there are new forms of distance learning, where students can count on multimedia content. In this work, a web-based platform has been developed, destined to distance learning. On the one hand, it allows a teacher to interact with a group of remote students simultaneously while visualizing in group DASH-based audiovisual content (AV) (for example, a video polimedia) on which the teacher can explain concepts teachers. On the other hand, students can also visualize in group the AV contents to be able to prepare these subjects in an interactive and collaborative way. As means of interaction, the platform includes text messaging tools, real-time audio and video communications, shared control of AV content playout, shared board and file sharing. Standard web-based technologies have been used for the development of the platform. So, compatibility between networks, between platforms and between devices is guaranteed. It also includes inter-stream synchronization solutions (AV content and text messages) and synchronization of the playout of the AV content between recipients (IDMS, Inter-destination Media Synchronization). The correct performance of the platform and the accuracy of the obtained synchronization have been evaluated objectively (QoS) and subjectively (QoE).

Keywords: e-learning, collaborative tools, interactivity, HTML5, DASH, WebRTC

RESUMEN

Actualmente, hay nuevas formas de enseñanza a distancia, donde los estudiantes disponen de contenido multimedia. En este trabajo se ha desarrollado una plataforma basada en web, destinada a la docencia a distancia. Por un lado, en la misma se permite que un docente interactúe con un grupo de alumnos remotos de forma simultánea mientras visualizan, en grupo, contenido audiovisual o AV basado en tecnología DASH (por ejemplo, un vídeo polimedia) sobre el que el docente puede explicar conceptos teóricos. Por otro lado, los alumnos también pueden visualizar en grupo los contenidos docentes disponibles en la misma para poder preparar dichas materias de forma interactiva y colaborativa. Como medios de interacción, la plataforma incluye herramientas de mensajería de texto, de comunicaciones de audio y vídeo en tiempo real, control compartido de la reproducción del contenido AV, pizarra compartida y de intercambio de ficheros. Para el desarrollo de la plataforma se han utilizado tecnologías estándar basadas en web, por lo que se garantiza la compatibilidad entre redes, entre plataformas y entre dispositivos. Además, incluye soluciones de sincronización inter-flujo (contenidos AV y mensajes de texto) y de sincronización de la reproducción de los contenidos AV entre destinatarios (IDMS, Inter-Destination Media Synchronisation). El correcto funcionamiento de la plataforma y la precisión de la sincronización obtenida se han evaluado de forma objetiva (QoS) y subjetiva (QoE).

Palabras Clave: Tele-enseñanza, herramientas colaborativas, interactividad, HTML5, DASH, WebRTC

Agradecimientos

Quiero agradecer a Fernando Boronat por haberme dado la oportunidad de realizar este trabajo junto a él y por haberme introducido en el mundo de la investigación.

También quiero agradecer a mi familia el apoyo y la confianza durante estos cuatro años.

Índice

1.	Introducción	6
1.1	Introducción.....	6
1.2	Objetivos	6
1.3	Estructura de la memoria	7
2.	Estado del arte.....	8
3.	Tecnologías utilizadas.....	10
3.1	Tecnologías web	10
3.1.1	HTML5	10
3.1.2	CSS3	10
3.1.3	JavaScript	10
3.1.4	AngularJS	11
3.1.5	Socket.IO	11
3.1.6	WebRTC	11
3.2	Node.js.....	11
3.3	JSON	12
3.4	MySQL	12
3.5	MPEG-DASH.....	12
4.	Plataforma desarrollada	13
4.1	Funcionalidades de la plataforma.....	13
4.1.1	Visión general	13
4.1.2	Subida de ficheros	14
4.1.3	Manejo de sesiones	15
4.1.3.1	Creación, unión y abandono de sesiones.....	15
4.1.3.2	Invitación a nuevos usuarios a participar en la sesión.....	16
4.1.4	Información de la sesión compartida	16
4.1.5	Consumo distribuido y sincronizado	16
4.1.5.1	Intercambio de roles de usuario	17
4.1.6	Herramientas de interacción y colaboración.....	17
4.1.6.1	Chat de texto.....	18
4.1.6.2	Pizarra compartida	18
4.1.6.3	Intercambio de ficheros.....	19
4.1.6.4	Audio/videoconferencia.....	19
4.2	Arquitectura y detalles de implementación	20
4.2.1	Arquitectura de la plataforma	20
4.2.1.1	Cliente.....	20

4.2.1.2	Servidores.....	22
4.2.2	Sincronización.....	23
4.2.2.1	Ajustes de configuración	23
4.2.2.2	Reloj virtual	24
4.2.2.3	Sincronización de reproducción de vídeo distribuido.....	26
4.2.2.4	Sincronización de chat privado de texto	29
4.2.3	Streaming de vídeo	30
4.2.4	Visualización de documentos PDF e imágenes.....	30
4.2.5	Pizarra compartida	30
4.2.6	Audio/Videoconferencia	33
5.	Evaluación	34
5.1	Evaluación objetiva	34
5.1.1	Objetivos de la evaluación	34
5.1.2	Escenario de evaluación	34
5.1.3	Resultados	36
5.1.4	Conclusiones evaluación objetiva de la plataforma	38
5.2	Evaluación subjetiva.....	38
5.2.1	Objetivos de la evaluación	38
5.2.2	Escenario de evaluación	39
5.2.3	Resultados	39
5.2.4	Conclusiones evaluación subjetiva de la plataforma	42
6.	Conclusiones y trabajo futuro	44
	Bibliografía.....	45
	Índice de figuras	46
	Índice de tablas.....	47

1. Introducción

1.1 Introducción

Tradicionalmente, el acceso a la educación ha sido mediante clases magistrales, en las que una persona (profesor) habla a una audiencia que escucha su explicación. Por ejemplo, en las clases de las universidades.

Actualmente, el paradigma ha cambiado radicalmente, ya que la educación tradicional se complementa o sustituye, en muchos casos, con elementos multimedia complementarios que los alumnos pueden reproducir dónde y cuándo quieran, además de poder hacerlo las veces que crean necesarias. Este método implica mucho cambio respecto al tradicional, ya que permite la educación a distancia y, además, el alumno es libre de ver el contenido cuando lo necesite y no ha de ajustarse a un horario concreto, como en la educación tradicional. Aun así, tiene el problema de que es unidireccional, ya que el alumno no tiene opción a interpelar o realizar preguntas al profesor.

Por otra parte, la educación a partir de elementos multimedia no permite, en un principio, que diferentes alumnos de un mismo contenido colaboren entre ellos si no es haciéndolo en el mismo lugar, de manera que uno de ellos reproduce el contenido multimedia y así poder comentarlo y ayudarse entre ellos.

La motivación de este Trabajo Final de Grado (TFG) es diseñar e implementar una plataforma web que permita la interacción, por un lado, entre alumnos y profesores y, por otro, entre alumnos geográficamente separados, en tiempo real, mientras, además, puedan estar visualizando un contenido multimedia de forma sincronizada. La plataforma se ha desarrollado mediante tecnologías estándares basadas en web (HTML5, CSS y JavaScript), lo que garantizará que sea multi-plataforma y multi-dispositivo. Otra ventaja de HTML5 es que solo es necesario un navegador web (no necesita ningún software específico) para la ejecución de la aplicación.

1.2 Objetivos

Los objetivos del TFG son:

- Diseño e implementación de una plataforma basada en tecnologías web de educación a distancia incluyendo el consumo de contenido multimedia sincronizado, además de la interacción entre los usuarios.
- Evaluación objetiva de la plataforma diseñada.

- Evaluación subjetiva de alumnos y profesores de la plataforma diseñada.

1.3 Estructura de la memoria

En el siguiente capítulo de Estado del arte se van a analizar y comparar aplicaciones existentes que ofrecen características similares a las de la plataforma presentada y que se utilizan en el campo de la enseñanza. A continuación, en el capítulo 3, se describen las tecnologías utilizadas para el desarrollo de la plataforma (tanto las del lado del servidor como las del lado del cliente). En el capítulo 4 se describe la plataforma desarrollada, primero se describen sus funcionalidades y, posteriormente, se detalla su implementación. En el capítulo 5 se evalúa la plataforma desarrollada, tanto de manera objetiva (QoS) como subjetiva (QoE). Por último, se presentan las conclusiones y el trabajo futuro en el capítulo 6.

2. Estado del arte

En esta sección se van a analizar dos plataformas que actualmente se utilizan para la enseñanza a distancia de manera interactiva para, posteriormente, compararlas con la plataforma presentada.

La primera plataforma es Adobe Connect, que es una aplicación comercial desarrollada por la empresa Adobe. La plataforma permite el uso de contenido compartido (documentos, imágenes, vídeos, ...) al mismo tiempo que se tiene comunicación en tiempo real a través de chat de texto y/o videoconferencia. También dispone de una pizarra donde el anfitrión puede dibujar, o escribir, y todos los usuarios de la sesión lo visualizan. Adobe Connect es una aplicación nativa, es decir, ha de ser instalada por el usuario, aunque también cuenta con una versión web para la cual no hace uso de tecnologías web, sino que utiliza Adobe Flash. Por lo tanto, para poder utilizar la plataforma en su versión web se ha de tener instalado Adobe Flash. Por otra parte, la pizarra compartida no permite visualizar el cursor, de manera que el anfitrión para señalar alguna cosa ha de dibujar algo (círculo, ...) en la pizarra.

La segunda plataforma es WebEx de Cisco. Esta plataforma también es una aplicación nativa, por lo que también ha de ser instalada por el usuario. Proporciona unas características similares a las de Adobe Connect, aunque sí que permite el uso del puntero compartido. En cuanto al resto de funcionalidades de pizarra compartida y videoconferencia es muy similar a Adobe Connect.

Estas dos aplicaciones han sido probadas en sistemas operativos Windows y Android. Al ser aplicaciones nativas, no son exactamente iguales en todos los sistemas operativos y en algunos casos no tienen las mismas funcionalidades. En el caso de Adobe Connect las aplicaciones sí que son muy similares en diseño y en funcionalidades. En cambio, la aplicación para Android de Cisco WebEx no permite el uso compartido de contenido siendo anfitrión, sí que permite recibir contenido compartido, pero no crearlo.

Estas dos plataformas están diseñadas para reuniones en línea y tareas de colaboración, aunque se utilizan en la actualidad para aplicaciones de tele-enseñanza. Tanto para Adobe Connect como para Cisco WebEx no está claro si se proporcionan ni cómo se proporciona la sincronización de flujos (incluyendo IDMS y la sincronización entre flujos), y si es así, los niveles de precisión que se alcanzan, ya que no se especifica en sus respectivas referencias.

En la Tabla 1 se puede ver un resumen de las funcionalidades de las plataformas analizadas y una comparativa con la plataforma presentada.

Tabla 1. Comparación entre la plataforma desarrollada y otras existentes

Funcionalidad	Adobe Connect	Cisco WebEx	Plataforma presentada
Grupos independientes	Si	Si	Si
IDMS	-	-	Si
Información de la sesión	Si	Si	Si
Cambio dinámico de roles	Si	Si	Si
Documentos compartidos	Si	Si	Si
Imágenes compartidas	Si	Si	Si
Vídeo compartido	Si	Si	Si
Pantalla compartida	Si	Si	No
Controles de reproducción compartidos	Si	Si	Si
Nivel de volumen compartidos	No	No	Si
Chat de texto privado	Si	Si	Si
Audio conferencia	Si	Si	Si
Audio/Video conferencia	Si	Si	Si
Puntero compartido	No	Si	Si
Pizarra compartida	Si	Si	Si+¹
Intercambio de ficheros	No	No	Si
Uso exclusivo de tecnología web	No	No	Si

[-] Significa que no se especifica en las fuentes

[+] Significa un rendimiento mejorado y/u opciones adicionales (ver notas al pie)

¹ Tanto Adobe Connect como Cisco WebEx permiten la pizarra compartida, pero los usuarios solo ven lo que el anfitrión ha dibujado cuando lo termina de dibujar, mientras que en la plataforma presentada se hace el dibujo en tiempo real.

3. Tecnologías utilizadas

3.1 Tecnologías web

3.1.1 HTML5

HTML (*HyperText Markup Language*) es un lenguaje de marcado que nos permite crear la estructura y el contenido web. Este lenguaje (basado en etiquetas) nos permite estructurar todo el contenido de la página web.

A partir de la última versión (HTML5) se permite la inclusión de contenido multimedia (audio y video) de manera nativa. Esta nueva funcionalidad es muy importante para el desarrollo de la presente plataforma ya que en versiones anteriores se tenían que incluir los elementos multimedia a través de tecnologías de terceros (por ejemplo, plugins Flash) lo cual creaba muchas incompatibilidades entre navegadores, además de la necesidad de que el usuario final tuviera dicha tecnología instalada.

3.1.2 CSS3

CSS es el lenguaje utilizado para definir el diseño del contenido definido en HTML. Tiene una sintaxis muy simple, solo es necesario tomar una referencia del elemento al que se le quiere otorgar el diseño y darle valor a cada una de las propiedades deseadas.

3.1.3 JavaScript

Se trata de un lenguaje de programación interpretado, que proporciona interactividad a los elementos de la página web. El hecho de que sea un lenguaje interpretado supone que no debe ser compilado para su distribución, pero necesita de un intérprete, que normalmente será el navegador que interpreta la página web (aunque puede hacerse también en la parte del servidor gracias a *Node.js*, como veremos más adelante).

Es un lenguaje basado en eventos, lo que tiene mucha importancia en la programación web ya que la mayoría de las funciones son ejecutadas por un evento producido por algún elemento de la página web.

Los elementos de HTML son tratados desde JavaScript gracias al DOM (*Document Object Model*) que es el punto de unión entre los elementos declarados en HTML y JavaScript y que permite la manipulación de dichos elementos, creando propiedades, métodos y eventos.

3.1.4 AngularJS

AngularJS es un *framework* de código abierto escrito en JavaScript creado por Google. Es utilizado principalmente para la creación de webs SPA (*Single-page Application*) que son aplicaciones web de una sola página la cual no se recarga cada vez que se actualiza el contenido, sino que carga toda la aplicación al iniciarla y solo actualiza el contenido de la página (si fuera necesario) evitando, de esta manera, la interpretación del código JavaScript en cada página.

Además, este *framework* hace uso de MVC (*Model View Controller*) el cual separa la parte del diseño (vistas del usuario) y de la lógica de negocio. De esta manera, se consigue un código mucho más ordenado y estructurado.

3.1.5 Socket.IO

Socket.IO es una librería escrita en JavaScript, que permite comunicación en tiempo real y bidireccional entre el navegador y un servidor web. Socket.IO maneja, de manera transparente, el proceso de conexión y abstrae los mecanismos de transporte. Utiliza el protocolo WebSocket, pero proporciona muchas otras funciones, como la difusión a múltiples sockets, el almacenamiento de datos asociados a cada cliente y el manejo de los eventos de manera asíncrona.

Las características principales de la librería Socket.IO son la fiabilidad, el soporte de reconexión automática, la detección de desconexión, el soporte binario (el cual permite transmitir cualquier estructura de datos), el soporte de multiplexación y el soporte de sala (formando grupos para el envío de mensajes a todo el grupo). Permite enviar diferentes tipos de datos a través del mismo canal de comunicación.

3.1.6 WebRTC

WebRTC es un proyecto abierto estandarizado por IETF (*Internet Engineering Task Force*) y W3C (*World Wide Web Consortium*) que permite a los navegadores y aplicaciones móviles la comunicación en tiempo real gracias a su API (*Application Programming Interface*). Permite el desarrollo de aplicaciones incluyendo la transmisión de audio y video en tiempo real entre distintos usuarios. También tiene un canal para la transmisión de datos (*DataChannel*)

3.2 Node.js

Node.js es un entorno de código abierto y multiplataforma, escrito en JavaScript, para aplicaciones del lado del servidor. Proporciona un modelo impulsado por eventos y una API de E/S (Entrada/Salida) sin bloqueo para el desarrollo de aplicaciones distribuidas,

eficientes y ligeras, con un uso intensivo de datos, al optimizar el rendimiento en términos de interactividad, escalabilidad, fiabilidad y rendimiento. Además, Node.js contiene una biblioteca incorporada para permitir que las aplicaciones actúen como servidor web sin la necesidad de software adicional (como, por ejemplo, Apache o Internet Information Services -IIS- Server).

3.3 JSON

JSON (*JavaScript Object Notation*) es un formato de texto para el intercambio ligero de datos. Es un subconjunto de la notación literal de los objetos en JavaScript (de ahí su nombre). Es considerado un lenguaje independiente debido a su amplia adopción como alternativa a XML.

3.4 MySQL

MySQL es un sistema de gestión de bases de datos relacional. Está considerado como uno de los gestores de bases de datos más popular del mundo, sobre todo para el desarrollo web.

3.5 MPEG-DASH

MPEG-DASH es una técnica de *streaming* adaptativo basado en HTTP. La técnica consiste en generar distintas copias de un mismo contenido con diferentes calidades y dividir cada copia en segmentos breves de corta duración. El cliente descargará y reproducirá los segmentos con una calidad mayor o menor, por ejemplo, según el ancho de banda y la disponibilidad de recursos del equipo (la lógica de decisión puede tener en cuenta muchos más parámetros del cliente).

4. Plataforma desarrollada

4.1 Funcionalidades de la plataforma

Las principales funciones de la plataforma desarrollada son las siguientes:

- Subida de contenidos al servidor multimedia de la plataforma
- Creación, unión y abandono a sesiones
- Invitación a nuevos usuarios a la sesión
- Información de la sesión actual (nombre de la sesión, participantes, nombre)
- Consumo multimedia de manera sincronizada
- Interacción en tiempo real mediante chat de texto, audio y videoconferencia, control compartido de la reproducción, pizarra compartida y envío de ficheros.

4.1.1 *Visión general*

Las figuras 1 y 2 muestran las páginas de inicio y principal de la plataforma, respectivamente. La página de inicio (Figura 1) incluye un menú superior con las opciones de crear sesiones, entrar a sesiones creadas y mostrar la información sobre la plataforma (versión, autores, ...). Además, se ha añadido una pequeña introducción con un breve tutorial sobre las funcionalidades de la plataforma. La Figura 2 presenta la página principal de la plataforma, que es la que los usuarios ven una vez que el usuario ya está participando en una sesión con otros usuarios. Todas las partes de la página se describen brevemente en cuadros de texto.



Figura 1. Página de inicio de la plataforma



Figura 2. Página principal de la plataforma

4.1.2 Subida de ficheros

La página de inicio de la plataforma (Figura 1) incluye una opción para subir archivos de vídeo al servidor multimedia de la plataforma, para tenerlos disponibles en las futuras sesiones. Cuando se presiona sobre el botón “Subir contenido”, se necesitan unas credenciales para poder acceder (parte izquierda de la Figura 3). Una vez autenticados, el usuario puede incluir un nombre al vídeo y seleccionar el archivo de vídeo a través del navegador (parte derecha de la Figura 3).

Figura 3. Subida de ficheros

Una vez subido el contenido, el servidor se encarga de generar el contenido en DASH y crear el *manifest*. Además de añadir en la base de datos el contenido generado.

4.1.3 Manejo de sesiones

4.1.3.1 Creación, unión y abandono de sesiones

Por un lado, los usuarios pueden crear sesiones nuevas presionando la opción *Crear una sesión* del menú superior de la página de inicio (Figura 1). Una vez presionado, aparece un formulario para crear la sesión. El usuario debe introducir el nombre de usuario y el nombre para la sesión. La parte izquierda de la Figura 4 presenta un ejemplo de un formulario completo con la creación de una sesión denominada “Tutoría” por el usuario “Fernando”. Al acabar de rellenar el formulario el usuario deberá presionar sobre el botón *Crear* y se abrirá la página principal de la plataforma (como la de la Figura 2, pero vacía, ya que ningún otro usuario se ha unido a la sesión y la reproducción del video aún no se habrá iniciado).

Figura 4. Página de creación (izquierda) y unión a la sesión (derecha)

Por otro lado, los usuarios pueden unirse a sesiones activas presionando el botón “Unirse a una sesión” del menú superior de la página principal de la plataforma (Figura 1). Al presionar el botón aparecerá un formulario vacío (parte derecha de la Figura 4) donde el usuario deberá introducir el nombre de la sesión, en ese momento se rellenará automáticamente el campo de participantes. Una vez introducido el nombre de usuario, podrá presionarse el botón “Unirse” para entrar la página principal de la plataforma (Figura 2). Cada vez que un usuario entra a una sesión, todos los demás miembros reciben una notificación a través del chat de texto.

Para ofrecer la posibilidad de abandonar la sesión, en el menú superior de la plataforma (Figura 2) existe la opción “Salir”. Cada vez que un usuario abandona una sesión, a todos los demás miembros también se les notifica a través del chat de texto.

4.1.3.2 *Invitación a nuevos usuarios a participar en la sesión*

En la parte superior derecha de la plataforma (Figura 2), junto con los iconos para activar o desactivar la videoconferencia aparece también un icono de compartir, que en el momento en que es presionado se copia en el portapapeles la URL de la sesión para ser enviada a otros usuarios vía, por ejemplo, email.

4.1.4 *Información de la sesión compartida*

En el menú superior de la aplicación (Figura 2) se puede observar la opción de “Información de la sesión”, si se presiona sobre ella se abre una ventana con toda la información referente a la sesión compartida, como el nombre de la sesión, el nombre de usuario, una lista de participantes y el rol del propio usuario y, también, el usuario que tenga el rol de maestro (Figura 5).



Figura 5. Bloque de información de la sesión

4.1.5 *Consumo distribuido y sincronizado*

Con el fin de permitir que los usuarios distribuidos consuman contenido multimedia de forma sincronizada se ha implementado un esquema Maestro/Esclavo para el control de la sesión y para el control de dicha sincronización.

Por un lado, un cliente es el cliente maestro y los otros clientes son los clientes esclavos. El maestro es el único usuario que puede ver y usar las opciones de control del contenido compartido, la reproducción del video (como reproducir / pausar, avance rápido, cambio de volumen y saltos) y de la pizarra compartida (Figura 6). Al principio, el usuario que creó la sesión será el maestro, pero los roles se pueden intercambiar, tal y como se explica en la siguiente subsección.



Figura 6. Controles del vídeo y activación de la pizarra compartida (Solo disponible para el usuario maestro)

La plataforma permite el consumo de diferentes contenidos compartidos, como documentos PDF, imágenes o videos. El maestro puede seleccionar entre los diferentes contenidos disponibles para compartirlo con todos los usuarios.

Para obtener la sincronización de todos los procesos de reproducción de video de los clientes, todos ellos comparten una referencia de tiempo común y se ha implementado una solución de sincronización entre destinatarios(IDMS) basándose en el esquema Maestro/Esclavo [1]. De esta manera, todos los clientes ven el mismo contenido de video en el mismo instante. Además, se ha implementado otra solución de sincronización Maestro/Esclavo para la sincronización entre flujos de video y secuencias de texto en cada cliente. Todos los mensajes de texto privados intercambiados tienen la marca de tiempo para ser presentados a todos los usuarios en el instante correcto para evitar inconsistencias durante la experiencia compartida. En otras palabras, la presentación de los mensajes está alineada en el tiempo con la posición asociada de la reproducción del video en cada uno de los clientes remotos.

4.1.5.1 Intercambio de roles de usuario

La plataforma incluye la posibilidad de intercambiar el rol de maestro entre los distintos usuarios. Cualquier usuario puede pedirle al maestro el intercambio de roles presionando sobre el botón “Ser Maestro” del menú superior de la plataforma (Figura 2). En ese momento le aparecerá al maestro un mensaje para aceptar o rechazar el cambio (Figura 7).



Figura 7. Notificación de cambio de maestro recibido

4.1.6 Herramientas de interacción y colaboración

En esta sección se describen las herramientas de interacción y colaboración incluidas en la plataforma: chat de texto, pizarra compartida, intercambio de ficheros y audio/videoconferencia.

4.1.6.1 Chat de texto

La plataforma incluye un chat de texto privado con baja latencia gracias a la conexión punto a punto de WebRTC (usando su canal de datos, *DataChannel*).

La Figura 8 muestra la interfaz del chat de texto. Como se puede observar, los mensajes aparecen en dos colores distintos destacando de esta manera los mensajes propios de los mensajes de otros usuarios.

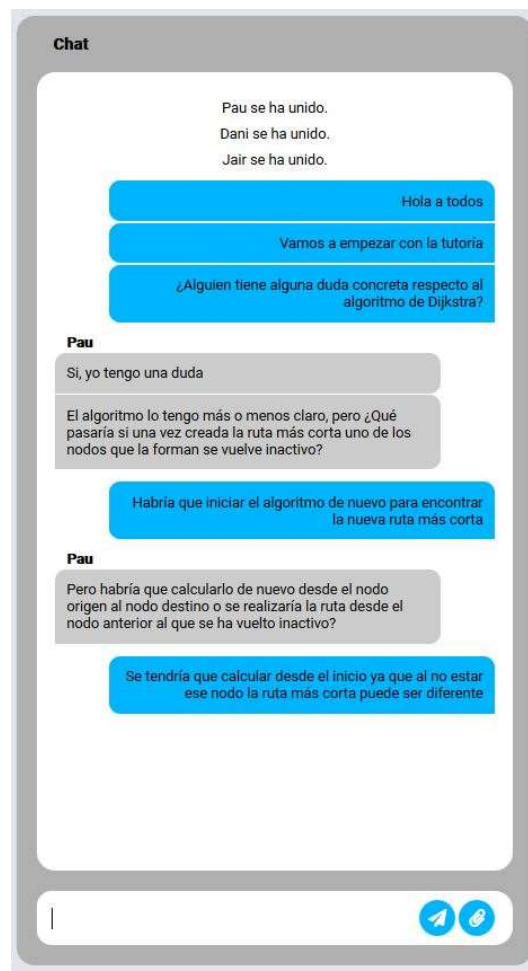


Figura 8. Chat de texto privado

4.1.6.2 Pizarra compartida

El usuario maestro puede compartir una pizarra con el resto de los usuarios. Cuando se activa la pizarra, automáticamente se pausa la reproducción del video (en caso de estar compartiendo un vídeo) y aparece una paleta con las opciones de la pizarra compartida (Figura 9). Dichas opciones consisten en dibujar a mano alzada, dibujar elipses, dibujar rectángulos, escribir texto, borrar todo el contenido de la pizarra o solo una parte. Además, permite cambiar el color y el tamaño del pincel o del texto. La pizarra también tiene la opción de puntero compartido, de manera que todos los esclavos ven el puntero del maestro.

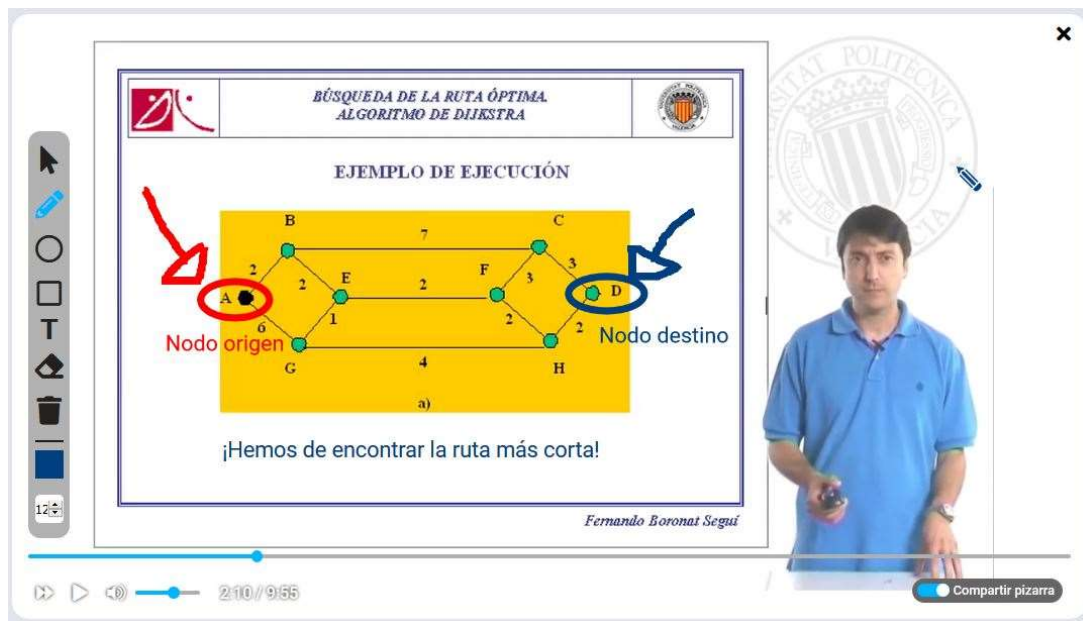


Figura 9. Ejemplo de pizarra compartida utilizada por el usuario maestro

4.1.6.3 Intercambio de ficheros

El chat privado incluye la posibilidad de compartir ficheros con el resto de los usuarios de la sesión. En la parte inferior derecha del chat de texto (Figura 8) aparece un botón para adjuntar un archivo a la conversación.

En el caso del envío de documentos PDF o imágenes a través del chat, estos ficheros podrán ser compartidos por el maestro con todos los usuarios de la sesión.

4.1.6.4 Audio/videoconferencia

La plataforma también tiene la opción de realizar audio/videoconferencia entre los distintos usuarios de una misma sesión. Según el estudio en [2] los usuarios prefieren sistemas de interacción basadas en texto, pero la audio/videoconferencia le aporta mayor interactividad y comunicaciones más confortables.

En la plataforma se inicia automáticamente la audio/videoconferencia entre los usuarios de la sesión, aunque cada usuario decide en qué momento envía su flujo de audio y/o video. Las cámaras web aparecen en la parte inferior de la plataforma y en el caso de que algún usuario tenga el micrófono activado, pero no tenga cámara, aparecerá un icono de una persona. Cada usuario puede mutear o desmutear a los demás usuarios presionando sobre el icono del altavoz de su cámara o icono (Figura 10).

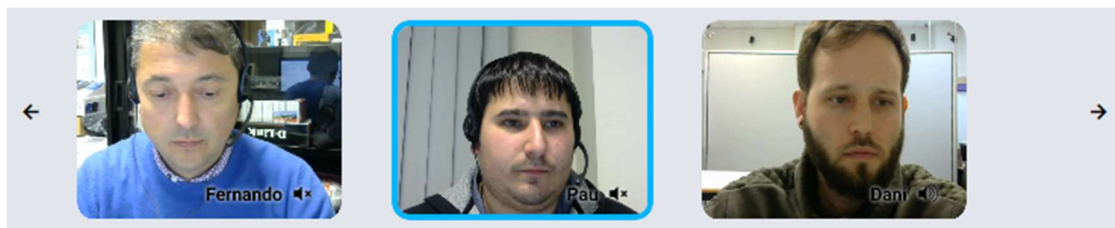


Figura 10. Audio y video conferencia

4.2 Arquitectura y detalles de implementación

En esta sección se comenta, primero, la arquitectura de la plataforma y, posteriormente, se presentan los detalles de implementación.

4.2.1 Arquitectura de la plataforma

La Figura 11 presenta la arquitectura general de la plataforma.

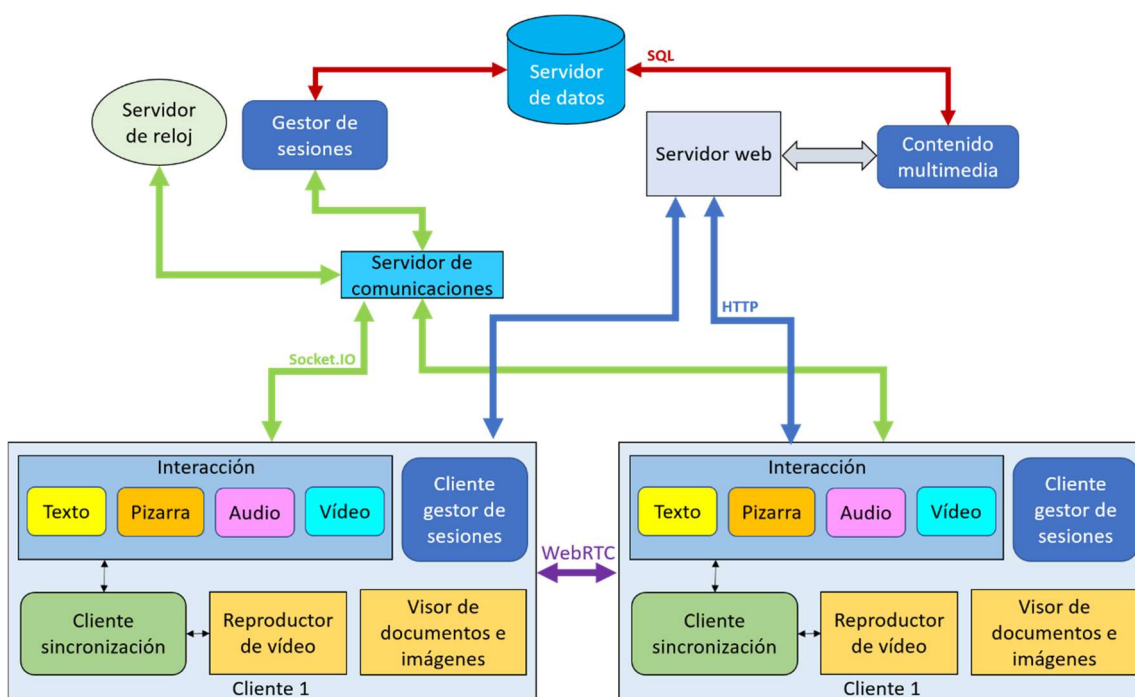


Figura 11. Arquitectura de la plataforma

4.2.1.1 Cliente

El cliente es el dispositivo utilizado por el usuario de la plataforma y que cuenta con ratón y teclado (o pantalla táctil, en su caso), cámara y micrófono. Es utilizado para acceder a través de un navegador a la plataforma.

La implementación del cliente se ha realizado utilizando el *framework* AngularJS. Se han creado distintos módulos para conseguir una plataforma totalmente modular, de la que, posteriormente, se pueda sacar una parte (por ejemplo, el reproductor de vídeo

sincronizado) y crear otra plataforma distinta, pero utilizando dicho módulo. Dispone de los siguientes módulos:

Cliente gestor de sesiones

El cliente gestor de sesiones es el encargado de crear, unirse o abandonar sesiones. Además, contiene toda la información de la sesión. Está conectado con el Gestor de sesiones a través del servidor de comunicaciones para mantener la sesión. También se encarga de crear las conexiones punto a punto de WebRTC.

Reproductor de vídeo

El reproductor de vídeo consiste en un cliente DASH que obtiene el contenido MPEG-DASH del servidor mediante peticiones HTTP GET.

Visor de documentos PDF e imágenes

El visor de documentos PDF e imágenes se encarga de obtener del servidor los documentos y las imágenes y mostrarlos en el cliente.

Cliente de sincronización

El cliente de sincronización es el módulo encargado de comunicarse con los demás módulos clientes de sincronización para sincronizar la reproducción del vídeo (IDMS). También controla la sincronización de los mensajes de chat con la reproducción del vídeo, a nivel local (inter-flujo).

Todos los mensajes de sincronización entre los clientes son transmitidos a través del *DataChannel* de WebRTC (Figura 12), proporcionando así unas comunicaciones de muy baja latencia comparado con otros sistemas que utilizan un servidor para todas estas comunicaciones, causando así una latencia mínima producida por el procesado de dicho servidor y el reenvío de los mensajes entre los distintos clientes.

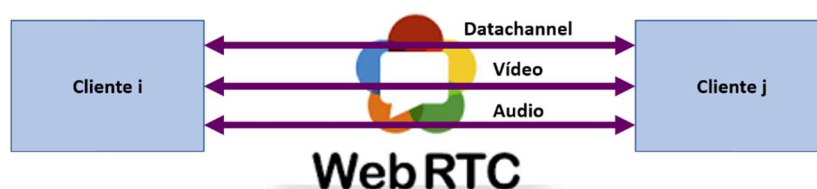


Figura 12. Canales de datos, audio y video bidireccionales entre clientes

Herramientas de interacción

Las herramientas de interacción se encargan de permitir la colaboración entre los diferentes usuarios a través del chat de texto, la audio/videoconferencia y la pizarra compartida.

Para estas comunicaciones también se utiliza la conexión WebRTC. La audio/videoconferencia utiliza los canales de audio y vídeo (RTP), mientras que la pizarra compartida y el chat de texto utilizan el canal de datos (*DataChannel*).

4.2.1.2 Servidores

Los servidores están implementados utilizando diferentes tecnologías (Apache, Node.js y MySQL). En esta versión de la plataforma está todo en una misma máquina, aunque podría estar distribuido en diferentes máquinas y comunicarse entre ellas.

Servidor Web.

El servidor web se encarga de almacenar los archivos de la plataforma y los recursos necesarios. También es el proveedor del contenido multimedia (DASH) de la plataforma. Está implementado con un servidor Apache.

Gestor de sesiones.

Se encarga de administrar las sesiones compartidas (creación, unión y abandono) en la plataforma. También proporciona todas las necesidades de señalización de las conexiones de WebRTC. Se comunica con el cliente gestor de sesiones de cada cliente gracias al servidor de comunicaciones.

Las comunicaciones bidireccionales y punto a punto de datos, audio y vídeo entre los distintos clientes se establecen utilizando la librería *RTCMultiConnection* de WebRTC, que permite las conexiones entre múltiples usuarios con WebRTC (Figura 12). Esta librería necesita un servidor de señalización que ha sido implementado en el gestor de sesiones (Figuras 11 y 13).

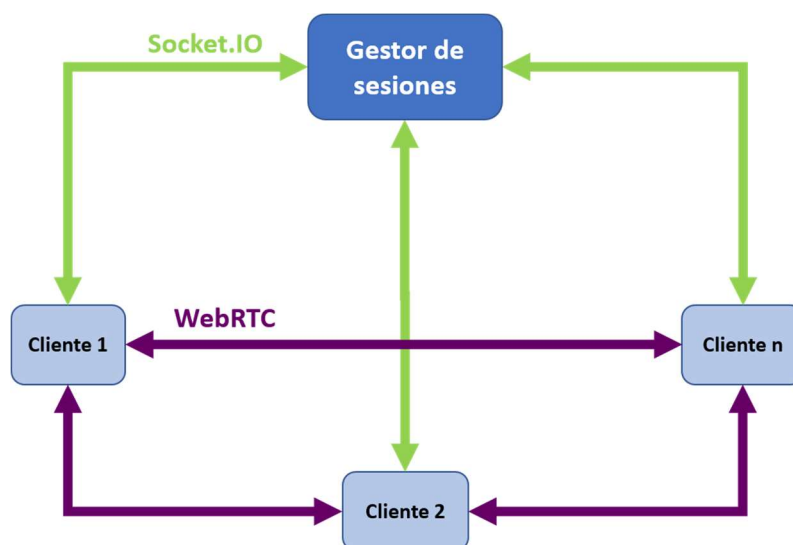


Figura 13. Intercambio de mensajes entre los clientes y el gestor de sesiones

Servidor de comunicaciones.

El servidor de comunicaciones es el encargado de proporcionar comunicación entre las diferentes entidades de la plataforma gracias a la librería Socket.IO (Utilizando la tecnología WebSocket).

Servidor de datos.

El servidor de datos es una base de datos MySQL que contiene la información de los contenidos multimedia y los datos de las sesiones y usuarios activos.

Servidor de reloj.

El servidor de reloj proporciona una referencia de reloj común a todos los clientes de sincronización.

4.2.2 Sincronización

La plataforma presentada requiere dos tipos de sincronización:

- 1) Entre elementos multimedia dentro del mismo dispositivo, conocida como sincronización inter-flujo.
- 2) Entre elementos multimedia siendo reproducidos en diferentes dispositivos y en diferentes puntos geográficos, conocida como IDMS (*Inter-Destination Media Synchronization*).

Para poder lograr los dos tipos de sincronización, se han implementado tres componentes en la plataforma: un reloj virtual, una solución de sincronización inter-flujo y una solución de sincronización IDMS.

4.2.2.1 Ajustes de configuración

Hay algunos parámetros (como la asincronía permitida, tiempo máximo de pausa permitido, intervalo de envío de mensajes IDMS...) que en la plataforma no se han dejado de manera fija, sino que se han puesto en un archivo de configuración para poder cambiarlas en un futuro, ya sea para labores de investigación (por ejemplo, prueba de nuevos algoritmos de sincronización) o para realizar cambios más fácilmente en la plataforma. Todos los parámetros definidos se resumen en la Tabla 2.

Tabla 2. Parámetros de configuración de sincronización

Parámetro	Descripción	Valor por defecto
Intervalo IDMS	Intervalo entre mensajes IDMS enviados por el maestro IDMS	5000 ms
Intervalo solicitud reloj virtual	Intervalo entre las solicitudes de los clientes al servidor de reloj virtual	10000 ms

Parámetro	Descripción	Valor por defecto
Umbral de asincronía	Valor de umbral permitido de la asincronía entre los procesos de reproducción del maestro y el esclavo IDMS	160 ms
Tipo de ajuste de asincronía	Tipo de ajuste de la asincronía que deben realizar los esclavos	'S&P' o 'AMP'
Porcentaje AMP permitido	Porcentaje en que se disminuye o aumenta la de velocidad de reproducción cuando se usa el ajuste del proceso de reproducción por AMP	0.25 (25%)
Tiempo máximo de asincronía AMP (TMA)	Asincronía máxima permitida para realizar el ajuste por AMP	5000ms
Tiempo máximo de pausa (TMP)	Tiempo máximo permitido para realizar una pausa	5000ms
Intervalo de sincronización inter-flujo	Intervalo de comprobación de la sincronización inter-flujo	100ms

4.2.2.2 Reloj virtual

La plataforma se basa en la existencia de líneas de tiempo virtuales y comunes para todas las entidades involucradas en una sesión con el fin de lograr la sincronización de los flujos. Se ha implementado un mecanismo de sincronización basado en un reloj virtual para obtener una referencia de reloj común entre los distintos clientes. Consiste en tener un servidor de reloj de referencia y un protocolo de solicitud-respuesta. A través del servidor de comunicaciones se intercambian unos mensajes periódicos con marcas de tiempo específicas entre los distintos clientes y el servidor de reloj para así estimar la diferencia entre los relojes, de manera similar a como lo hace NTP (*Network Time Protocol*, RFC 5906 [3]). El mecanismo se basa en tener en cuenta la diferencia de tiempo que existe entre el reloj local y el reloj común sin modificar el reloj local. Este servidor no es un servidor dedicado, sino que puede estar junto con otros servidores (Figura 11).

La Figura 14 muestra el proceso de intercambio de mensajes entre el cliente (C) y el servidor de reloj (S) en el algoritmo de sincronización de reloj virtual propuesto. En primer lugar, el cliente envía un mensaje tipo *Request* donde indica t_1 (hora local del cliente en el instante de envío). Cuando el servidor recibe el mensaje responde con un mensaje tipo *Reply* donde devuelve el t_1 enviado por el cliente y también t_2 (hora local del servidor en el instante de llegada del mensaje) y t_3 (hora local del servidor en el instante de salida del mensaje). Se incluyen los dos instantes de tiempo para así poder tener una mayor precisión, teniendo en cuenta el tiempo de procesado del servidor. Cuando este mensaje llega al cliente, el cliente registra t_4 (hora local del instante de llegada del mensaje). Con estos cuatro instantes de tiempo, el cliente puede medir el retardo de ida y vuelta de la red (RTT, del inglés Round-Trip-Time) utilizando la ecuación

(1) y también el tiempo de ida del mensaje (asumiendo redes simétricas, es decir, que el tiempo de ida y el tiempo de vuelta son iguales) con la ecuación (2).

$$RTT_{C \rightarrow S} = (t_4 - t_1) - (t_3 - t_2) \quad (1)$$

$$d_{C \rightarrow S} = \frac{RTT_{C \rightarrow S}}{2} = \frac{(t_4 - t_1) - (t_3 - t_2)}{2} \quad (2)$$

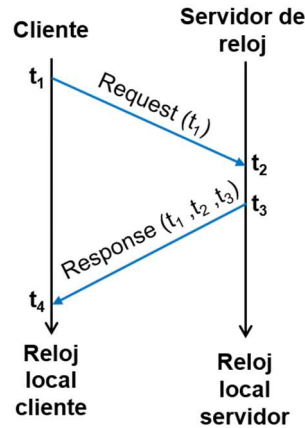


Figura 14. Intercambio de mensajes para la estimación de latencia y las diferencias de reloj

Además, es posible estimar la diferencia (offset) entre el reloj del cliente y el reloj del servidor mediante la ecuación (3). Ejecutando de manera periódica este algoritmo es posible estimar y compensar el offset entre los relojes del cliente y el servidor y de esta manera proporcionar una noción de tiempo igual para todos los clientes. Los clientes obtienen el instante actual del reloj virtual utilizando la ecuación (4).

$$offset_{C \rightarrow S} = t_2 - t_1 - \frac{(t_4 - t_1) - (t_3 - t_2)}{2} = t_2 - t_1 - d_{C \rightarrow S} \quad (3)$$

$$Reloj\ virtual\ cliente = Reloj\ local\ cliente + offset_{C \rightarrow S} \quad (4)$$

Este mecanismo, conocido comúnmente como sincronización de reloj virtual ([4]), permite alinear los relojes de los clientes con el reloj del servidor de reloj sin la necesidad de instalación de complementos o módulos adicionales ni la utilización de la misma tecnología de sincronización de reloj.

El intervalo de sincronización (periodo entre envíos de preguntas del cliente al servidor) puede ser un valor de tiempo fijo (por ejemplo, 5 o 10s) o ser variable según el número de clientes (para no sobrecargar el servidor o no exceder el ancho de banda de este). La estimación regular del offset del reloj permitirá actualizar continuamente dichos valores y hacer frente a las fluctuaciones temporales.

4.2.2.3 Sincronización de reproducción de vídeo distribuido

En sesiones compartidas de vídeo donde diferentes usuarios está visualizando el mismo contenido de vídeo y, además, están interactuando entre ellos en tiempo real (videoconferencia o chat de texto) es necesaria una sincronización de la reproducción del vídeo ya que si no se hace puede llegar a ser molesto para los usuarios si la asincronía supera los 400ms ([5]). Por lo tanto, el objetivo de la plataforma en cuanto a sincronización es conseguir no superar esta asincronía entre los distintos clientes IDMS de una misma sesión.

En la plataforma se ha implementado un esquema Maestro/Esclavo (M/S) de sincronización ([1]) para cada sesión (puede haber varias simultáneamente y cada una tendrá su mecanismo de sincronización independiente), donde uno de los usuarios es el maestro IDMS y los demás serán los esclavos IDMS. El usuario maestro IDMS coincidirá siempre con el maestro de la sesión, es decir, el usuario que tiene el control de la reproducción del vídeo, de la pizarra, etc.

El maestro IDMS de la sesión envía periódicamente mensajes de control IDMS a todos los demás participantes de la sesión compartida. Se envían a través del canal de datos, *DataChannel*, de WebRTC (Figura 15). El intervalo de envío de mensajes puede ser, al igual que en el caso de los mensajes del reloj virtual, fijo o variable dependiendo del número de clientes de la sesión para no sobrecargar el ancho de banda de la red.

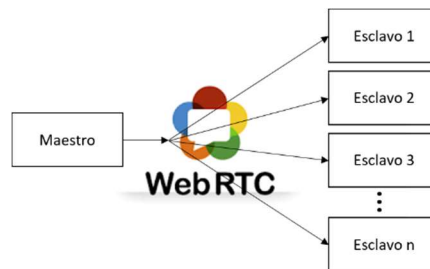


Figura 15. Intercambio de mensajes IDMS vía WebRTC

El usuario maestro incluye en los mensajes IDMS la siguiente información: el instante actual de reproducción del vídeo y el instante de reloj virtual en el que se ha tomado la posición del vídeo. Cada cliente IDMS, cuando recibe un mensaje IDMS utiliza la ecuación (5) para calcular la asincronía entre su reproducción de vídeo y la del maestro.

$$\begin{aligned} \text{Asincronía} &= \text{PosRef} - \text{PosCliente} + \Delta_{\text{Relojes}} \\ &= \text{PosRef} - \text{PosCliente} \text{ RelojVirtual} - \text{RelojRef} \end{aligned} \quad (5)$$

, donde *PosRef* es la posición de reproducción enviada por el maestro, *PosCliente* es la posición actual de reproducción propia, *RelojVirtual* es el instante actual del reloj virtual y *RelojRef* es el instante de reloj virtual recibido por el maestro. Si la asincronía es positiva significa que la reproducción del cliente IDMS va retrasada respecto a la del maestro IDMS, en cambio, si es negativa significa que va adelantada respecto a la del maestro IDMS.

En caso que la asincronía calculada exceda un umbral permitido, el esclavo debe ajustar su proceso de reproducción para sincronizarse con el maestro. El instante de reloj es necesario incluirlo en el mensaje IDMS para poder tener en cuenta el retardo de la red a la hora de calcular la asincronía.

Los ajustes del proceso de reproducción en los clientes IDMS pueden ser corregidos de dos maneras distintas, utilizando ajustes agresivos (saltos y pausas) o suavizados (AMP, *Adaptive Media Playout*, [6]).

Cada vez que se recibe un mensaje IDMS del maestro, se calcula la asincronía. Cuando el valor absoluto de la asincronía supera un cierto umbral, se ha de iniciar el proceso de sincronización correspondiente, que por defecto es AMP. Como el ajuste de la asincronía mediante AMP no es inmediato, si la asincronía es muy alta ($> TMA$) se procedería a hacer el ajuste por saltos y pausas (útil cuando entra un cliente con la sesión ya iniciada).

El esquema M/S de sincronización IDMS presenta un inconveniente muy claro, el abandono del maestro IDMS de la sesión, ya sea voluntariamente o no. Este problema se ha solucionado aplicando la siguiente política de re-elección de maestro: de los usuarios de la sesión se le asignará el rol de maestro al usuario más antiguo, es decir, al primero que haya entrado en la sesión.

Ajuste de la asincronía mediante saltos y pausas

La técnica de la corrección de manera agresiva consiste en realizar saltos o pausas de la magnitud de la asincronía, de manera que si se tiene una asincronía positiva se ha de hacer un salto hacia delante de la duración de asincronía, en cambio, si la asincronía es negativa se ha de realizar una pausa durante un período de tiempo igual a la asincronía.

Para evitar pausas muy largas, si se supera un valor determinado de asincronía negativa (TMP) se realizará un salto hacia atrás para corregir la asincronía.

Ajuste de la asincronía mediante AMP

La corrección de la asincronía de manera suavizada (AMP) consiste en aumentar o reducir la tasa de reproducción del vídeo durante el intervalo de tiempo necesario para sincronizarse con el maestro IDMS lo antes posible. Esta técnica es la más conveniente ya que no realiza cambios bruscos en la reproducción del vídeo y, por tanto, es menos agresiva para el usuario. Según [7], si el cambio de la tasa de reproducción es inferior al 25% no es perceptible por el usuario y dependiendo del contenido hasta el 50% podría llegar a ser aceptable.

El ajuste por AMP se basa en acelerar (si la asincronía es positiva) o decelerar (si la asincronía es negativa) el proceso de reproducción del vídeo durante un tiempo determinado haciendo uso de la ecuación (6).

$$\text{Tiempo de ajuste} = \left| \frac{\text{asincronía}}{\text{porcentaje AMP}} \right| \quad (6)$$

Después de este tiempo de ajuste la tasa de reproducción se restaurará a su valor nominal.

Intercambio de mensajes de sincronización

Para poder ofrecer interactividad a la sesión, el maestro IDMS, además del mensaje de sincronismo IDMS, también comparte los comandos del control de la reproducción (play, pause y cambio de posición) para, por ejemplo, pausar en un determinado instante del vídeo para resolver alguna duda de algunos de los alumnos asistentes a la sesión o para volver a ver alguna parte del vídeo.

Por una parte, en el caso de la acción “play” se envía el instante del vídeo y además una referencia de reloj, para que, de esta manera, los usuarios empiecen a reproducir en el instante actual de reproducción del maestro IDMS (teniendo en cuenta el retardo de la red). Por otra parte, si la acción es un cambio de posición (*seek*) se envía un mensaje con la nueva posición, pero no se inicia la reproducción. Los esclavos IDMS al recibir este mensaje cambian a la posición enviada por el maestro IDMS y, en cuanto se hayan descargado el suficiente contenido para empezar a reproducir, enviarán un mensaje al maestro IDMS informando de que ya pueden iniciar la reproducción. El maestro IDMS, al recibir este mensaje de todos los esclavos IDMS, empezará la reproducción y enviará un mensaje de “play” a todos ellos.

Para compartir toda esta información se han definido diferentes mensajes que el maestro IDMS envía a los esclavos IDMS. Todos los mensajes relacionados con el

control de la reproducción, la sincronización IDMS y el reloj virtual se han resumido en la Tabla 3.

Tabla 3. Mensajes de control para lograr la sincronización

Propósito	Dirección	Tipo	Campos	Descripción
IDMS	M → S	Tiempo de reproducción	PosRef	Posición del vídeo
			RelojRef	Instante de obtención de la posición (referente al reloj virtual común)
Control de reproducción	M → S	Play	Posición	Posición del vídeo en el que se va a empezar a reproducir
			Reloj	Instante en el que se inicia la reproducción (referente al reloj virtual común)
		Pausa	Posición	Posición del vídeo en el que se realiza la pausa
		Cambio de posición	Posición	Posición del vídeo del nuevo punto de reproducción
	S → M	Puedo reproducir	-	
Sincronización del reloj virtual	C → S	<i>Request</i> reloj	t1	Instante de reloj local del cliente en el que envía la petición
	S → C	<i>Reply</i> reloj	t1	Instante del reloj local del cliente en el que ha enviado la petición al servidor.
			t2	Instante del reloj local del servidor en el que recibe la petición del cliente.
			t3	Instante del reloj local del servidor en el que envía la respuesta al cliente

M: Maestro IDMS; S: Esclavo IDMS; S: Servidor de reloj; C: Cliente (Maestro y esclavo)

4.2.2.4 Sincronización de chat privado de texto

Los mensajes de texto se envían entre los clientes a través del *DataChannel* de WebRTC. Para evitar que un mensaje se muestre antes que el contenido al que se refiere dicho mensaje haya sido visto por el resto de los clientes, se ha implementado un mecanismo de sincronización de los mensajes de texto con la reproducción del vídeo. Para esta sincronización se ha implementado, también, un mecanismo M/S (Maestro/Esclavo), donde el flujo de vídeo es el flujo maestro y el flujo de texto es el flujo esclavo. Los mensajes de chat, además del texto del mensaje escrito por el usuario también incluyen la posición de reproducción del vídeo en el que se han enviado. Cuando otro cliente recibe un mensaje de chat comprueba si la posición de vídeo indicada en dicho mensaje es superior a la posición de la reproducción de su vídeo. Si lo es, muestra el mensaje de texto, y, si no lo es, deberá esperarse un cierto tiempo (por ejemplo, 100ms) y volver a hacer la comprobación.

En caso de no tener contenido compartido entre los usuarios o que el contenido compartido no sea de vídeo (es decir, imágenes o documentos PDF) los mensajes se muestran directamente.

4.2.3 Streaming de vídeo

El *streaming* del vídeo se realiza con la técnica MPEG-DASH para evitar problemas por poco ancho de banda de los usuarios de la sesión. Esta técnica consiste en tener alojado en el servidor el mismo contenido de vídeo en diferentes calidades, de esta manera el cliente elige si reproduce contenido con mayor o menor calidad teniendo en cuenta el ancho de banda disponible y, en algunos casos, los recursos disponibles en el equipo.

4.2.4 Visualización de documentos PDF e imágenes

Para la visualización de documentos PDF se utiliza un elemento canvas de HTML5. Cuando el maestro comparte un archivo PDF envía a los demás usuarios de la sesión la dirección de dicho fichero y todos lo muestran renderizando la primera página en el lienzo del canvas.

Además del fichero seleccionado, el maestro ha de enviar a los esclavos los movimientos realizados en el documento (*scroll* del documento y cambio de página) para que todos los usuarios visualicen la misma parte del documento. En la Tabla 4 se detallan los mensajes enviados por el usuario maestro a los esclavos.

Tabla 4. Mensajes/eventos relacionados con la pizarra compartida

Tipo	Campo	Valor
Fichero	Fichero	Ruta del fichero compartido
Scroll	ScrollTop	Scroll superior del documento (En %)
Cambio de página	Página	Número de la nueva página

4.2.5 Pizarra compartida

Para implementar la pizarra compartida se utilizan tres elementos *canvas* de HTML5, que sirven para dibujar objetos, líneas o texto sobre un lienzo. El primero de ellos se utiliza para mostrar el dibujo definitivo, el segundo se utiliza para mostrar el objeto que se está dibujando (elipses o rectángulos) y, por último, el tercer canvas se utiliza para mostrar el puntero, el lápiz de dibujo o la goma para borrar.

Para que todos los usuarios vean el mismo dibujo independientemente del tamaño de su pantalla, un elemento canvas tiene dos resoluciones distintas: la del lienzo (común a

todos los usuarios de la misma sesión) y la resolución en la que se muestra el dibujo (que depende de la pantalla de cada usuario).

Para el procesado de la pizarra se utilizan los eventos *pointerdown*, *pointermove* y *pointerup* que indican cuándo se ha presionado con el puntero (engloba a ratón de escritorio, acciones táctiles en pantallas que lo permitan o la utilización de lápices), cuándo se ha movido el puntero y cuándo se ha soltado el puntero, respetivamente.

Cuando el maestro entra el puntero en la pizarra compartida (si está activada) se empiezan a escuchar los tres eventos. Los eventos *pointerdown* y *pointerup* sirven para saber el estado del puntero (presionado o no, en caso de ratones de escritorio; o en contacto con la pantalla o no, en pantallas táctiles).

Según la herramienta seleccionada, los eventos ejecutan unas u otras acciones, aunque en todos los casos se dibuja un cursor (cursor, lápiz, goma, o cursor de texto) en la posición correspondiente.

Compartición del cursor

Si la función activada es la compartición de cursor los eventos *pointerdown* y *pointerup* no realizan ninguna acción. En este caso, el evento *pointermove* solo debe dibujar el cursor en la posición correspondiente.

Dibujo a mano alzada

En el caso del dibujo a mano alzada, si el puntero está presionado (ha ocurrido en evento *pointerdown*, pero todavía no *pointerup*), cada vez que se dispare el evento *pointermove* se dibujará una línea para unir el punto anterior el actual.

Como se puede observar, el dibujo a mano alzada en realidad se realiza con una sucesión de líneas rectas, pero la resolución que nos ofrece el evento *pointermove* es más que suficiente para realizar el dibujo a mano alzada a base de líneas rectas.

En esta utilidad el cursor es el lápiz.

Dibujo de elipses o rectángulos

En el caso del dibujo de elipses o rectángulos se utilizan dos *canvas*, el de dibujo final y el de figuras. En este caso, mientras el puntero está presionado, cada vez que se dispara el evento *pointermove*, primero se elimina el contenido del *canvas* de figuras y luego se dibuja la figura correspondiente (elipse o rectángulo) en él, teniendo en cuenta la posición donde se ha pulsado el cursor y la posición actual (esquina superior izquierda y esquina inferior derecha).

Por último, cuando se suelta el cursor (*pointerup*) se elimina la figura del *canvas* de figuras y se dibuja en el *canvas* del dibujo final.

Este proceso es necesario ya que el elemento *canvas* de HTML5 es una matriz de puntos (píxeles) y no reconoce objetos. Por tanto, no se puede eliminar un objeto del lienzo, sino que se han de eliminar sus píxeles. Esto provoca que si durante el transcurso del dibujo de una figura, dicha figura se posicionara por encima de otro objeto dibujado con anterioridad al borrar esta figura borraríamos también el objeto anterior. Por este motivo, las figuras generadas durante el acto de dibujar se realizan en un lienzo y una vez se terminan de dibujar se realizan en el lienzo definitivo.

Mientras está activa la función de dibujo de elipses o rectángulos el cursor, al igual que en el caso del dibujo a mano alzada, es el lápiz.

Escritura de texto

En el caso de tener activada la función de escribir texto en el momento en el que se cliquea el puntero (*pointerdown + pointerup*) se creará un campo de texto en la posición correspondiente para escribir el texto deseado. En el momento de volver a clicar el puntero o cuando se desactiva la función de texto el contenido del campo de texto se ha de escribir en el *canvas* del dibujo final.

En esta funcionalidad el cursor es un cursor de texto.

Borrado parcial

La función de borrado parcial es la más compleja en cuanto a su implementación. Al igual que ocurre a la hora de hacer dibujo a mano alzada se ha de borrar formando líneas rectas entre los puntos que ofrece el evento *pointermove*, pero para borrar contenido de un elemento *canvas* se ha de hacer siempre en forma de rectángulo, por tanto, hemos de encontrar la línea recta entre los dos puntos para ir borrando, de manera manual, rectángulos siguiendo la línea.

Mientras está activa la función de borrar, el cursor es un cuadrado blanco, representando a una goma.

Borrado total

La función de borrado total borra todo el contenido del *canvas* del dibujo final. El borrado total no es una función de la pizarra en sí. Ya que es un acto, no tiene cursor propio.

Intercambio de mensajes de la pizarra

Para compartir todas estas acciones con el resto de usuario de la sesión se han creado una serie de mensajes que el maestro envía a los esclavos a través del *DataChannel* de la conexión WebRTC para que reproduzcan sus movimientos sobre su pizarra. En la Tabla 5 aparecen todos los mensajes enviados por el maestro con sus respectivos campos y valores.

Tabla 5. Mensajes/eventos relacionados con la pizarra compartida

Tipo	Campo	Valor
Activar/desactivar pizarra	ActivarPizarra	True/false
Pointerdown	x	Coordenada X
	y	Coordenada Y
Pointermove	x	Coordenada X
	y	Coordenada Y
Pointerup	--	--
Cambiar función	Dibujando	'Cursor', 'Linea', 'Circulo', 'Rectangulo', 'Texto' o 'Goma'
Borrar	--	--
Cambio texto	Texto	Texto actual del campo de texto
Cambio color	Color	Nuevo color (formato: '#RRGGBB')
Cambio grosor	Grosor	Nuevo grosor

4.2.6 Audio/Videoconferencia

Cuando un usuario activa el envío de audio y/o vídeo se inicia la transmisión del flujo correspondiente (audio y/o video) por los canales de la conexión WebRTC destinados a ello (Figura 12).

Al iniciar la transmisión del flujo multimedia, los demás usuarios deben mostrar el vídeo o en su defecto un icono representando al usuario que tiene activo el micrófono e identificarlo con el nombre de usuario y añadir un botón para poder mutear, en caso de desearlo, el micrófono del usuario remoto.

5. Evaluación

En este capítulo se presentan los resultados de las evaluaciones realizadas sobre la plataforma (tanto objetivas como subjetivas). Primero, se presentan los resultados de la evaluación objetiva para comprobar el rendimiento de la plataforma desarrollada y comprobar el buen funcionamiento del algoritmo de sincronización. Posteriormente, se presentan los resultados obtenidos en las evaluaciones subjetivas realizadas con usuarios para comprobar la facilidad y comodidad de uso.

5.1 Evaluación objetiva

5.1.1 *Objetivos de la evaluación*

Para la evaluación de la calidad de servicio (QoS) de la plataforma se han considerado los siguientes aspectos:

- Rendimiento de la técnica de sincronización de vídeo (IDMS)
- Compatibilidad de la plataforma para distintos dispositivos y sistemas operativos
- Compatibilidad de la plataforma para distintos navegadores web

5.1.2 *Escenario de evaluación*

Para evaluar de manera objetiva la plataforma se ha diseñado una red con 5 equipos clientes y el servidor en un entorno de red de área local (LAN), pero se ha simulado una red de área amplia (WAN) creando retardos y pérdidas entre los equipos. La arquitectura de la red se puede ver en la Figura 16. Se puede observar la arquitectura creada para la simulación de una red WAN en un entorno controlado. Se puede observar cómo se han creado 4 redes independientes para los clientes a las que se les ha añadido un retardo de 100 ± 40 ms y unas pérdidas de paquetes de un 0.1%. En cada una de las redes hay un equipo de escritorio y en una de las redes se ha instalado un punto de acceso que proporciona conectividad a un dispositivo portátil inalámbrico (Tablet). Además, hay una red independiente en la que se ha instalado el servidor de la plataforma.

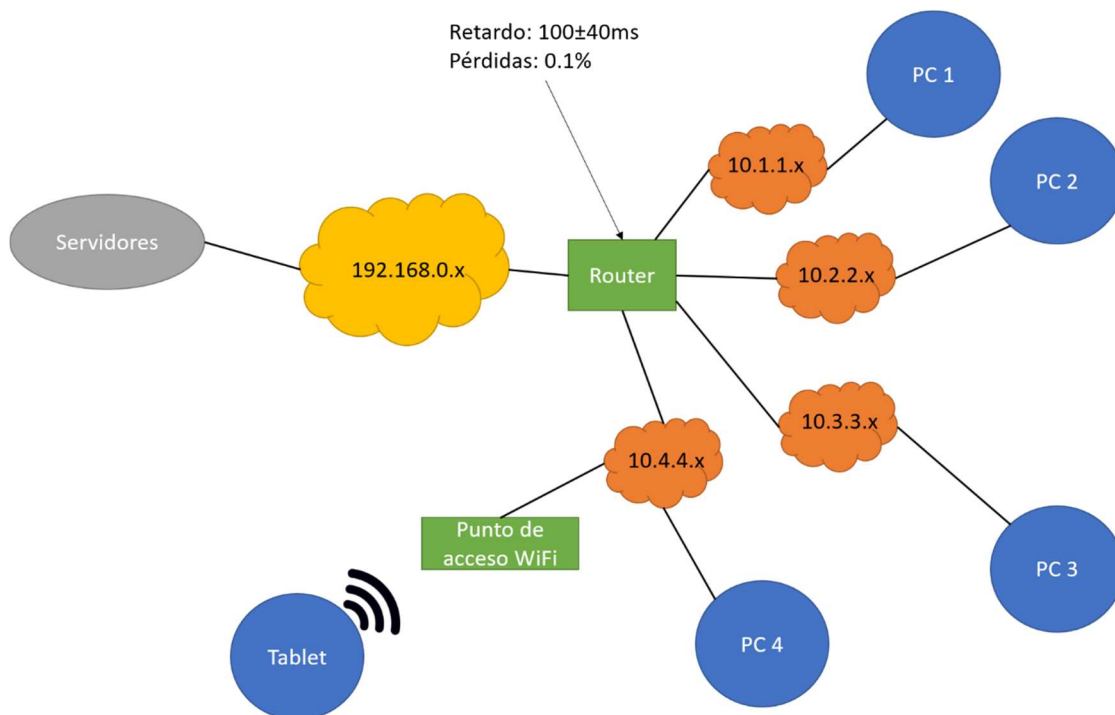


Figura 16. Arquitectura de las mediciones de la evaluación objetiva

Según [8] las desviaciones de los relojes de distintos equipos, sobre todo los de bajo coste, varían entre 10 y 100 ppm (partes por millón). Para realizar las evaluaciones se ha supuesto el peor caso (100ppm, que equivale a $\pm 0.0001\%$) y se le ha aplicado esta diferencia a dos de los equipos (PC2 y PC3) de manera positiva y negativa, respectivamente.

De esta manera, se puede comprobar cómo se comportaría la solución de sincronización en el peor de los casos. En la Tabla 6 se pueden ver los distintos equipos con su sistema operativo, navegador utilizado y la tasa de reproducción utilizada.

Tabla 6. Equipos utilizados en las evaluaciones objetivas

Equipo	Sistema Operativo	Navegador	CPU	RAM	Tasa de reproducción
PC1	Windows 10	Google Chrome	Quad-Core, 3.4GHz	8GB	1
PC2	Windows 10	Google Chrome	Quad-Core, 3.4GHz	8GB	1.0001
PC3	Windows 10	Google Chrome	Quad-Core, 3.4GHz	8GB	0.9999
PC4	Windows 10	Google Chrome	Quad-Core, 3.4GHz	8GB	1
Dispositivo inalámbrico	Android 8	Google Chrome	Quad-Core, 2.15GHz	4GB	1

5.1.3 Resultados

La primera prueba se ha realizado con los cinco equipos clientes con el algoritmo de sincronización implementado poniendo un vídeo a reproducir sin interacción entre los usuarios ni cambios (play, pause, cambio de posición) en la reproducción del vídeo.

En la Figura 17 se puede observar los resultados. En el equipo PC2 (que tiene la tasa reproducción superior a la normal) se puede observar como la asincronía va disminuyendo con el paso del tiempo, en el equipo PC3 (que se le ha rebajado la tasa de reproducción) se puede comprobar cómo la asincronía va aumentando con el paso de tiempo. Además, también se puede observar que en el equipo portátil (Tablet) también va aumentando la asincronía sin haber modificado su tasa de reproducción, esto demuestra que los relojes de los equipos, sobre todo, los portátiles tienen desviaciones en cuanto a su tasa de reloj y es necesaria su corrección.

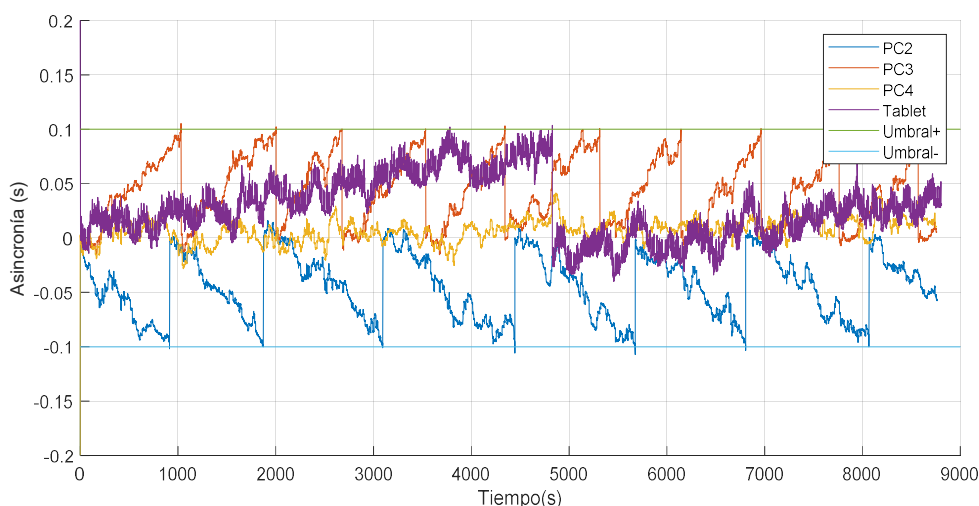


Figura 17. Asincronías prueba 1

En la Figura 17 se puede observar cómo tras 30 minutos aproximadamente en la sesión se superan los límites de asincronía de 400ms. Esto demuestra que, para poder ofrecer en la plataforma contenidos de mayor duración, es necesaria la implementación de algún algoritmo de sincronización IDMS.

Los umbrales del algoritmo de sincronización se han establecido en ± 100 ms con respecto al maestro IDMS. De esta manera, la asincronía máxima (entre el más adelantado y el más atrasado) debería ser inferior a 200ms, dejando un margen para los 400ms que no se deben superar para mantener una buena QoE.

En la Figura 17 se pueden observar el número de sincronizaciones realizadas a lo largo de una sesión. Cabe destacar también el ajuste realizado por la Tablet ya que en este

equipo no ha sido modificada la tasa de reproducción, de manera que, si no se hubiera realizado la sincronización, el usuario de ese equipo habría notado la asincronía y le hubiera causado molestias, degradando, así, su QoE.

Por último, se ha realizado una prueba en la que se han efectuado cambios en el proceso de reproducción del vídeo (pausas, saltos, ...) por parte del usuario maestro, y, además, se ha activado y desactivado la videoconferencia (activación de webcams y envío de audio y vídeo) de algunos usuarios, simulando, de esta forma, una sesión normal entre distintos alumnos y/o profesores.

En la Figura 18 se puede observar una parte de la sesión simulada entre distintos usuarios, en la que se han realizado distintas acciones sobre el vídeo y sobre la sesión. Se puede observar cómo los equipos de escritorio (PC2, PC3 y PC4) soportan de manera aceptable diferentes acciones (ver Tabla 7). En cambio, en el equipo portátil (Tablet) se puede observar como en cada acción se produce una asincronía (llegando en algunos casos a valores cercanos a 1 segundo) que debe ser corregida para la correcta visualización del contenido de manera sincronizada con el resto de los usuarios. En la Tabla 7 se pueden observar las acciones realizadas sobre la sesión y los respectivos tiempos en los que se producen.

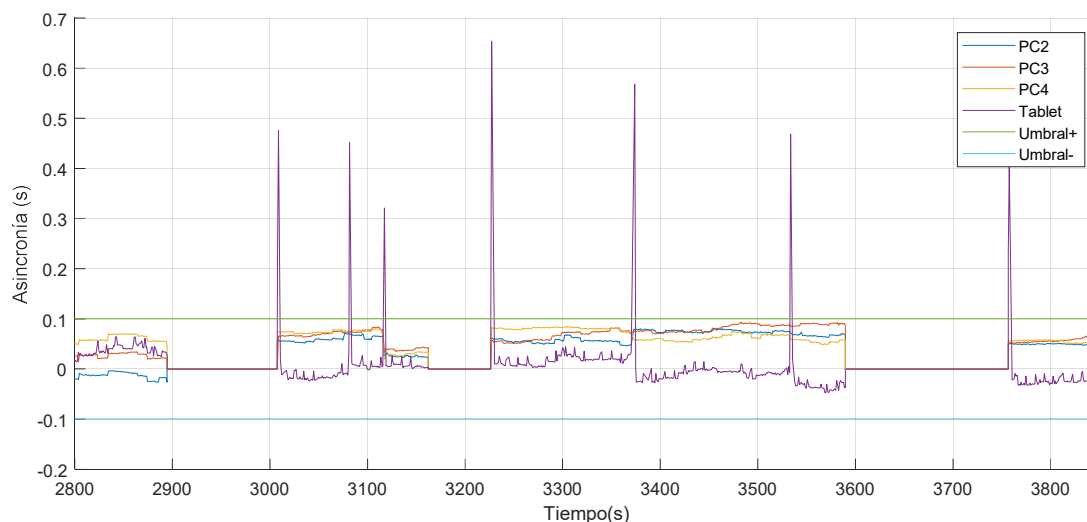


Figura 18. Sesión simulada

Tabla 7. Acciones de la sesión simulada

Tiempo (s)	Acción
2898	Pausa del vídeo
2898	Inicio de la pizarra compartida

Tiempo (s)	Acción
3005	Play del vídeo
3017 - 3070	Conversación en el chat de texto
3076	Usuario activa videoconferencia
3115	Salto del punto de reproducción del vídeo
3172	Pausa de vídeo por inicio de la pizarra compartida
3225	Play del vídeo
3240	Usuario desactiva videoconferencia
3368	Salto del punto de reproducción del vídeo
3529	Usuario activa videoconferencia
3686	Pausa de vídeo por inicio de la pizarra compartida
3753	Play del vídeo
3820	Usuario desactiva videoconferencia

5.1.4 Conclusiones evaluación objetiva de la plataforma

En los resultados obtenidos se puede observar que es necesario el ajuste de las asincronías debido a imperfecciones de los relojes y de las tasas de reproducción o a acciones realizadas por los usuarios de la sesión.

También se ha podido comprobar que el algoritmo de sincronización implementado funciona de manera correcta corrigiendo aquellas asincronías que superan el umbral permitido configurado.

5.2 Evaluación subjetiva

5.2.1 Objetivos de la evaluación

Para la evaluación de la calidad de experiencia (QoE) se han considerado los siguientes aspectos:

- Experiencia. Se pretende conocer la experiencia previa de los usuarios en plataformas similares, para conocer si han utilizado alguna herramienta que ofrezca características similares a la plataforma presentada.
- Usabilidad. Se pretende determinar el SUS (*System Usability Scale*) de la aplicación, realizando las preguntas concretas para poder determinarlo.

- Interactividad. Se pretende saber la opinión de los usuarios respecto a las diferentes herramientas de interacción de la plataforma.
- Feedback. Se plantean preguntas abiertas para obtener la opinión de los usuarios sobre la plataforma y posibles mejoras que puedan aportar. También se pregunta sobre la aplicabilidad actual de la aplicación.

5.2.2 Escenario de evaluación

El escenario para la evaluación subjetiva de la plataforma se ha realizado en un entorno real. Los servidores se han instalado en un servidor Cloud de la empresa clouding.io y en un servidor interno de la universidad y los usuarios han accedido a la plataforma desde sus equipos personales escogiendo ellos mismos la ubicación de la prueba (en casa, en la universidad, ...).

Realizando las pruebas de esta manera se consiguen unos resultados más realistas, ya que los usuarios se encuentran en un ámbito más privado y personal y no en un laboratorio. Además, no es necesario simular redes WAN, tal y como se hizo en las pruebas objetivas, ya que las pruebas se realizan en redes WAN reales.

Se han encuestado 12 alumnos de la universidad y de otros estudios superiores (Conservatorio Superior de Música de Alicante).

5.2.3 Resultados

En primer lugar, se van a ver los resultados en cuanto a experiencia de los usuarios en aplicaciones con características similares a la desarrollada. En la Figura 19 se puede observar que solo el 33.3% ha utilizado plataformas similares con anterioridad.

¿Utilizas herramientas telemáticas para realizar reuniones/tutorías en grupo?

12 respuestas

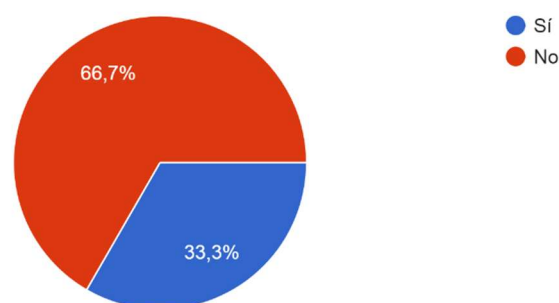


Figura 19. Experiencia previa de los usuarios

En cuanto a las herramientas disponibles en las plataformas utilizadas por los usuarios con anterioridad, en la Figura 20 se puede observar las herramientas disponibles en dichas plataformas. Se puede destacar que ninguna de las herramientas utilizadas por los encuestados utiliza pizarra compartida.

Indica las funcionalidades de interacción que utilizas de dichas herramientas

4 respuestas

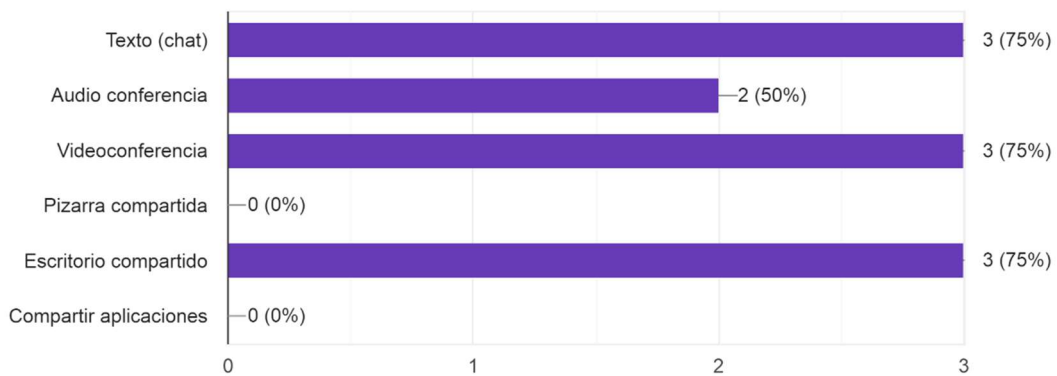


Figura 20. Herramientas de las plataformas utilizadas por los usuarios

En segundo lugar, se presentan los resultados obtenidos del test SUS, que valora la usabilidad de la plataforma. En dicho test, se ha obtenido una puntuación de 83.36 sobre 100. Se puede observar que la plataforma ha sido muy bien valorada por los usuarios en cuanto a utilidad y facilidad de uso.

En tercer lugar, se procede a ver los resultados en cuanto a interactividad de la plataforma. En la Figura 21 se puede observar el interés de los diferentes encuestados respecto a los diferentes canales de interacción disponibles en la plataforma. Se puede ver que el canal preferido por los usuarios es la videoconferencia respecto a otros canales como audioconferencia o chat de texto.

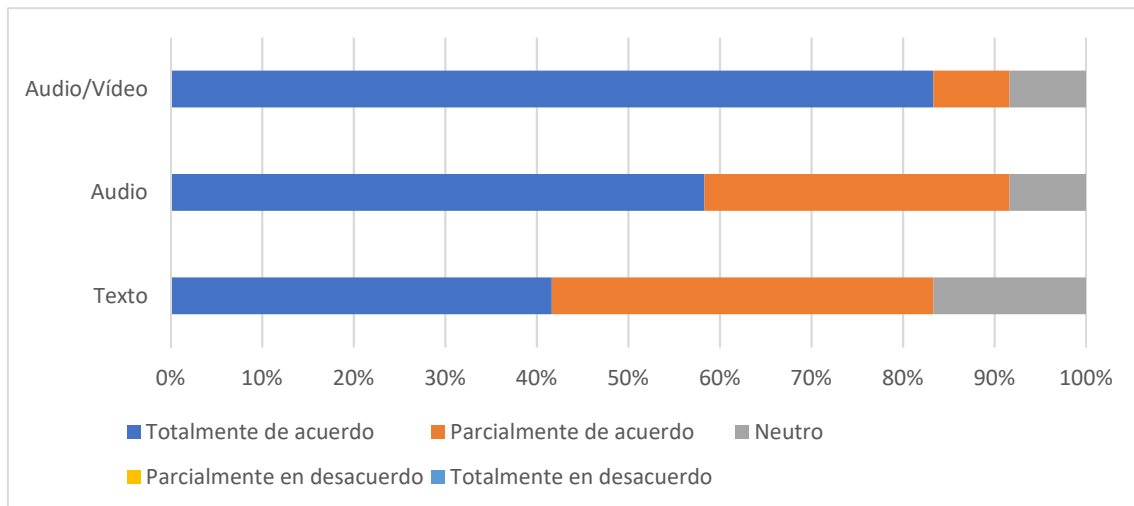


Figura 21. Opinión respecto a los canales de interacción

En la Figura 22 se puede observar la opinión de los encuestados respecto a las diferentes herramientas de plataforma. Se puede observar que las herramientas preferidas por los usuarios son la pizarra y el puntero compartido y las menos preferidas el control compartido de volumen y la conmutación de roles durante la sesión.

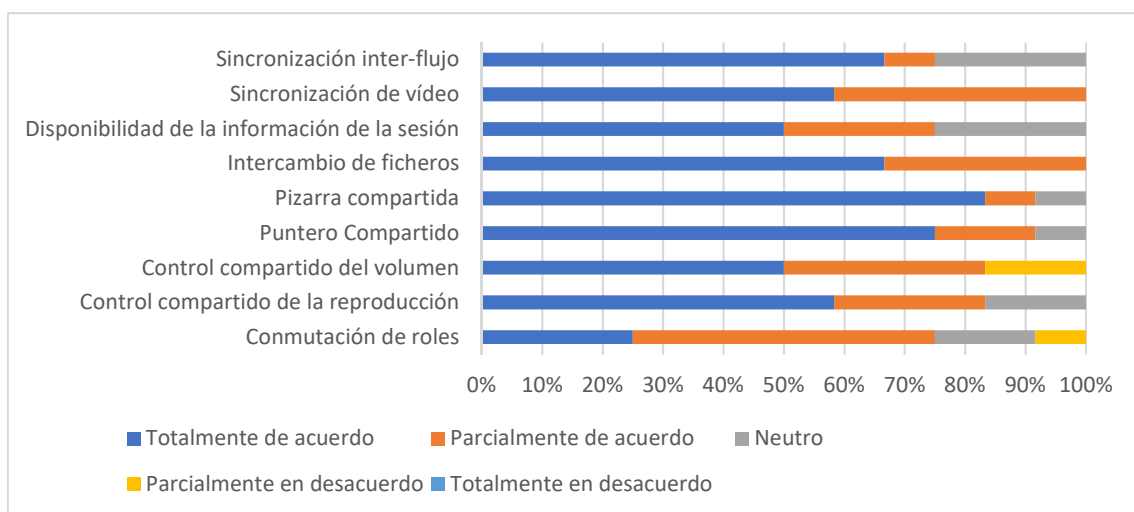


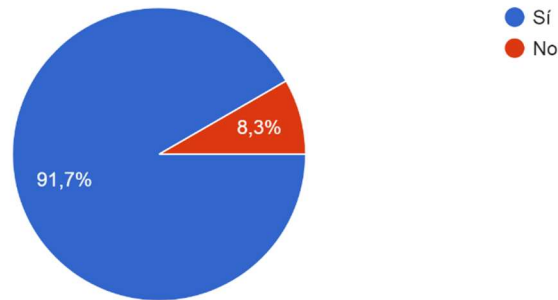
Figura 22. Utilidades de la plataforma

Por último, se muestran los resultados por lo que respecta al feedback ofrecido por los usuarios. En este apartado se pregunta por el interés en la utilización de la plataforma y el impacto que creen que pueda tener. Además, también se pregunta sobre mejoras que creen necesarias en la plataforma.

En la Figura 23 se puede observar como el 91.7% de los usuarios utilizaría la plataforma en caso de estar disponible. Además, el 63.7% creen que tendría un gran impacto en el paradigma actual y/o futuro de la docencia.

¿Utilizarías esta plataforma si estuviese disponible online?

12 respuestas



Esta plataforma puede tener un impacto en el paradigma actual y/o futuro en la docencia

11 respuestas

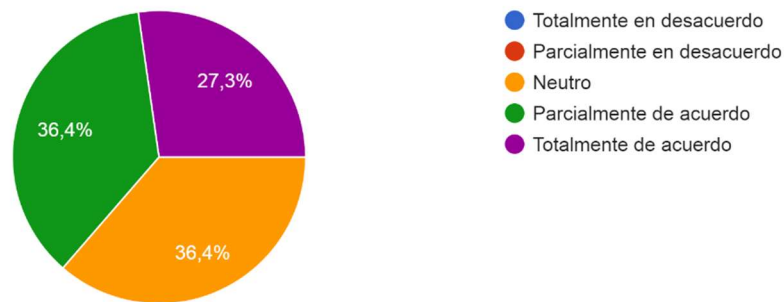


Figura 23. Feedback de los usuarios

Respecto a las mejoras propuestas por los usuarios se destacan:

- La implementación de una interfaz de usuario más colorida
- La opción de compartir pantalla por parte del maestro

5.2.4 Conclusiones evaluación subjetiva de la plataforma

Con las respuestas de los usuarios se puede concluir que la plataforma presentada causaría un gran impacto en el entorno docente de la actualidad, siendo utilizada por la mayoría de los usuarios.

Para la interacción durante las sesiones, los usuarios prefieren utilizar canales de vídeo y audio frente a canales de texto o solamente audio.

Plataforma web multimedia para docencia interactiva y colaborativa a distancia

Por lo que respecta a las herramientas que ofrece la plataforma, los usuarios las valoran muy positivamente, destacando sobre todo la pizarra compartida.

6. Conclusiones y trabajo futuro

Como conclusiones, en primer lugar, se considera que se han logrado los objetivos inicialmente planteados en el TFG. Por un lado, se ha creado una plataforma para la realización de reuniones telemáticas grupales entre alumnos y/o profesores, con la posibilidad de realizar videoconferencia entre los distintos asistentes, además de un chat de texto y una pizarra compartida utilizando solamente tecnología estándares web y consiguiendo, de esta manera, crear una plataforma multi-dispositivo y multi-plataforma. Por otro lado, se ha comprobado, primero mediante evaluaciones objetivas, el correcto funcionamiento de los protocolos, técnicas y algoritmos de sincronización implementadas y posteriormente, mediante evaluaciones subjetivas, la facilidad y comodidad de uso, además de la opinión de alumnos y profesores de la necesidad de una plataforma con las características de la plataforma desarrollada en el presente trabajo.

Como trabajo futuro, se pretende realizar la implementación de nuevas funcionalidades (algunas de ellas propuestas por los usuarios) a la plataforma:

- Posibilidad de reproducción de vídeos externos al servidor propio de la plataforma, por ejemplo, YouTube.
- Permitir el uso de otros tipos de contenidos como, por ejemplo, documentos de texto, presentaciones de diapositivas, ...
- Posibilidad de compartir la pantalla por parte del maestro.
- Mejoras en la interfaz gráfica de la plataforma.
- Integración con sistemas de usuarios como, por ejemplo, los usuarios de la UPV e incluso integrarlo dentro de la plataforma PoliformaT.
- Creación de sesiones programadas con antelación.
- Grabación de las sesiones

Bibliografía

- [1] F. Boronat, J. Lloret, and M. García, "Multimedia group and inter-stream synchronization techniques: A comparative study," *Inf. Syst.*, vol. 34, no. 1, pp. 108–131, 2009.
- [2] F. Boronat, M. Montagud, D. Marfil, and C. Luzon, "Hybrid Broadcast/Broadband TV Services and Media Synchronization: Demands, Preferences and Expectations of Spanish Consumers," *IEEE Trans. Broadcast.*, vol. 64, no. 1, pp. 52–69, 2018.
- [3] D. Millis, J. Martin, J. Burbank, and W. Kasch, "Network Time Protocol Version 4. Protocol and Algorithms." 2010.
- [4] H. Melvin, J. Shannon, and K. Stanton, *Time, frequency and phase synchronisation for multimedia- Basics, issues, developments and opportunities*. Springer International Publishing, 2018.
- [5] B. Rainer, S. Petscharnig, C. Timmerer, and H. Hellwagner, "Is one second enough? Evaluating QoE for inter-destination multimedia synchronization using human computation and crowdsourcing," in *2015 7th International Workshop on Quality of Multimedia Experience, QoMEX 2015*, 2015.
- [6] M. Montagud, F. Boronat, B. Roig, and A. Sapena, "How to perform AMP? Cubic adjustments for improving the QoE," *Comput. Commun.*, vol. 103, pp. 61–73, 2017.
- [7] S. Park and J. Kim, "An adaptive media playout for intra-media synchronization of networked-video applications," *J. Vis. Commun. Image Represent.*, vol. 19, no. 2, pp. 106–120, 2008.
- [8] F. Ferrari, A. Meier, and L. Thiele, "Accurate Clock Models for Simulating Wireless Sensor Networks," in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, 2010, p. 21:1--21:4.

Índice de figuras

Figura 1. Página de inicio de la plataforma	14
Figura 2. Página principal de la plataforma	14
Figura 3. Subida de ficheros	15
Figura 4. Página de creación (izquierda) y unión a la sesión (derecha)	15
Figura 5. Bloque de información de la sesión	16
Figura 6. Controles del vídeo y activación de la pizarra compartida (Solo disponible para el usuario maestro)	17
Figura 7. Notificación de cambio de maestro recibido	17
Figura 8. Chat de texto privado	18
Figura 9. Ejemplo de pizarra compartida utilizada por el usuario maestro.....	19
Figura 10. Audio y video conferencia	20
Figura 11. Arquitectura de la plataforma	20
Figura 12. Canales de datos, audio y video bidireccionales entre clientes	21
Figura 13. Intercambio de mensajes entre los clientes y el gestor de sesiones	22
Figura 14. Intercambio de mensajes para la estimación de latencia y las diferencias de reloj.....	25
Figura 15. Intercambio de mensajes IDMS vía WebRTC	26
Figura 16. Arquitectura de las mediciones de la evaluación objetiva.....	35
Figura 17. Asincronías prueba 1	36
Figura 18. Sesión simulada.....	37
Figura 19. Experiencia previa de los usuarios.....	39
Figura 20. Herramientas de las plataformas utilizadas por los usuarios	40
Figura 21. Opinión respecto a los canales de interacción	41
Figura 22. Utilidades de la plataforma.....	41
Figura 23. Feedback de los usuarios	42

Índice de tablas

Tabla 1. Comparación entre la plataforma desarrollada y otras existentes	9
Tabla 2. Parámetros de configuración de sincronización	23
Tabla 3. Mensajes de control para lograr la sincronización	29
Tabla 4. Mensajes/eventos relacionados con la pizarra compartida.....	30
Tabla 5. Mensajes/eventos relacionados con la pizarra compartida.....	33
Tabla 6. Equipos utilizados en las evaluaciones objetivas	35
Tabla 7. Acciones de la sesión simulada	37