



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

EVALUACIÓN DE WEBRTC EN REDES MÓVILES

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Ernesto Tomás Verdú

Tutor: Pietro Manzoni

2016-2017

Resumen

En este proyecto se ha desarrollado una aplicación para móviles Android que permite el envío de un vídeo en tiempo real a una plataforma web accesible por cualquier usuario. Mediante la geolocalización y un mapa en la web, los usuarios pueden acceder a las emisiones de vídeo que les interese. Se ha evaluado las prestaciones de dicha plataforma en redes móviles.

Palabras clave: WebRTC, Angular, Aplicación móvil, Ionic.

Abstract

This project focussed on the development of an application for Android phones that allows the streaming of video in real time to web platform accessible by any user. Using geolocation and a map in the web, users can access the video streams that interests them. The performance of this platform has been evaluated in mobile networks.

Keywords : WebRTC, Angular, App, Ionic.

Tabla de contenidos

1. Introducción	11
Motivación	11
Objetivo	11
2. Estado del arte	12
Youtube	12
Periscope	12
Facebook	13
La aplicación móvil	14
La página web	14
3. Metodología de desarrollo.....	15
Historias de usuario	15
Pruebas de aceptación	16
Kanban	17
4. Diseño	18
Tecnologías utilizadas	18
HTML.....	18
CSS	18
Bootstrap	18
TypeScript.....	18
Angular	19
Ionic	19
Node.js	19
WebRTC.....	19
Crosswalk.....	21
Firebase.....	21
Arquitectura de WebRTC.....	22
Servidor de señalización.....	23
Arquitectura de Angular	25
Módulo.....	25
Componente	26



Plantilla.....	26
Metadatos	26
Data Binding.....	26
Directiva.....	26
Servicio	26
La inyección de dependencias	26
Historias de usuario	27
Enviar vídeo	27
Visualizar mapa	28
Crear configuración	28
Ver marcadores.....	29
Mostrar vídeo.....	29
Ver datos del vídeo	30
Registro usuario.....	31
Autenticar	31
Lista de los vídeos en directo	32
Realizar búsqueda	33
Restablecer contraseña.....	33
Gestión de usuarios	34
5. Desarrollo	35
Interfaz de usuario	35
Registro de usuarios	35
Acceso	38
Restablecer contraseña.....	38
Crear una emisión en directo	41
La página web	44
6. Evaluación de las prestaciones.....	47
7. Conclusiones.....	51
Ampliaciones futuras	51
8. Bibliografía	52
Anexo A. Firebase.....	53
Anexo B. Estructura del proyecto	54

Tabla de ilustraciones

Ilustración 1: Youtube en directo	12
Ilustración 2: Periscope versión web	13
Ilustración 3: Facebook live	13
Ilustración 4: Historia de usuario	16
Ilustración 5: Kanban.....	17
Ilustración 6: Navegadores web.....	20
Ilustración 7: Navegadores web para móvil.....	20
Ilustración 8: Encontrar candidatos en la conexión.....	23
Ilustración 9: Mensajes en la base de datos de Firebase.....	23
Ilustración 10: Conexión entre dos candidatos	24
Ilustración 11: Arquitectura de Angular.....	25
Ilustración 12: Pantalla de acceso	35
Ilustración 13: Mensaje correo incorrecto	36
Ilustración 14: Dirección ya registrada	36
Ilustración 15: Longitud mínima de la contraseña.....	36
Ilustración 16: Registro completado	37
Ilustración 17: Correo electrónico.....	37
Ilustración 18: Contenido del correo	37
Ilustración 19: Mensaje de verificación	38
Ilustración 20: Dirección de correo no registrada.....	38
Ilustración 21: Contraseña incorrecta.....	38
Ilustración 22: Error al restablecer la contraseña	39
Ilustración 23: Restablecer contraseña	39
Ilustración 24: Correo recibido	39
Ilustración 25: Contenido del correo	40
Ilustración 26: Pantalla para restablecer la contraseña	40
Ilustración 27: Pantalla principal	41
Ilustración 28: Configuración	42
Ilustración 29: Desplegable.....	42
Ilustración 30: Emisión en directo	43
Ilustración 31: Página principal	44



Ilustración 32: Mapa con marcadores	44
Ilustración 33: Marcador seleccionado.....	45
Ilustración 34: Filtro de búsqueda.....	45
Ilustración 35: Desplegable temática.....	46
Ilustración 36: Transferencia emisor.....	47
Ilustración 37: Ancho de banda receptor.....	48
Ilustración 38: Parámetros audio	48
Ilustración 39: Parámetros vídeo.....	49
Ilustración 40: Grafica audio	50
Ilustración 41: Grafica vídeo	50
Ilustración 42: Firebase	53
Ilustración 43: Usuarios Firebase.....	53

1. Introducción

Actualmente los dispositivos móviles han evolucionado en cuanto a prestaciones y uso, las funciones como la realización de llamadas y el envío de mensajes han pasado a un segundo plano. Las tarifas de datos cada vez son más económicas, permitiendo el acceso a internet a velocidades aceptables sin pensar en sobrecostes, el estar siempre conectado posibilita la comunicación en tiempo real.

La utilización del móvil para mostrar que se hace en un determinado momento es muy habitual, a la gente le gusta y quiere estar conectado para ver que hacen sus allegados o gente que sigue o le interesa. Las redes sociales hacen esta función, pero con las emisiones de vídeo en directo existe la posibilidad de tener una comunicación directa, hay un contacto directo entre personas que pueden compartir información en tiempo real.

Motivación

El motivo de la elección de este TFG fue porque me pareció muy interesante la utilización de tecnologías en tiempo real ya que cada día está más en auge, ya sea para enviar vídeo, enviar audio, compartir ficheros, juegos, etc.

Otro motivo es la utilización de tecnologías que no había utilizado hasta el momento, el poder aprender y usar tecnologías que se usan en el mercado laboral y la combinación de desarrollar una aplicación móvil y una plataforma web suponía un desafío al que no me había enfrentado nunca.

Objetivo

El objetivo de este proyecto es desarrollar una aplicación para móviles Android que permita el envío de un vídeo en tiempo real a una página web accesible por cualquier usuario. El vídeo dispondrá de información de geolocalización de tal manera que el usuario podrá buscar en un mapa una determinada ubicación y seleccionar la emisión que más le interese.

El envío del vídeo debe ser de usuario a usuario sin utilizar un servidor de vídeo, dado que se pretende tener el máximo rendimiento posible. Se pretende evaluar las prestaciones que ofrece la plataforma web.



2. Estado del arte

Previo a todas las etapas del desarrollo del software se ha realizado una búsqueda de información y tecnologías.

Dada la existencia de diferentes plataformas, se ha observado que, en las páginas web existentes para ver transmisiones en directo no se permite la posibilidad de elegir fácilmente lo que se quiere ver y no existe la posibilidad de seleccionar una localización en concreto sin la necesidad de estar registrado. A modo de ejemplo:

Youtube

Youtube en directo permite ver los vídeos por temáticas, se puede buscar por un usuario, pero no dispone de un buscador o un mapa para seleccionar una emisión en directo de un determinado lugar. Se echa en falta una forma fácil de acceder a los vídeos de una cierta ubicación.

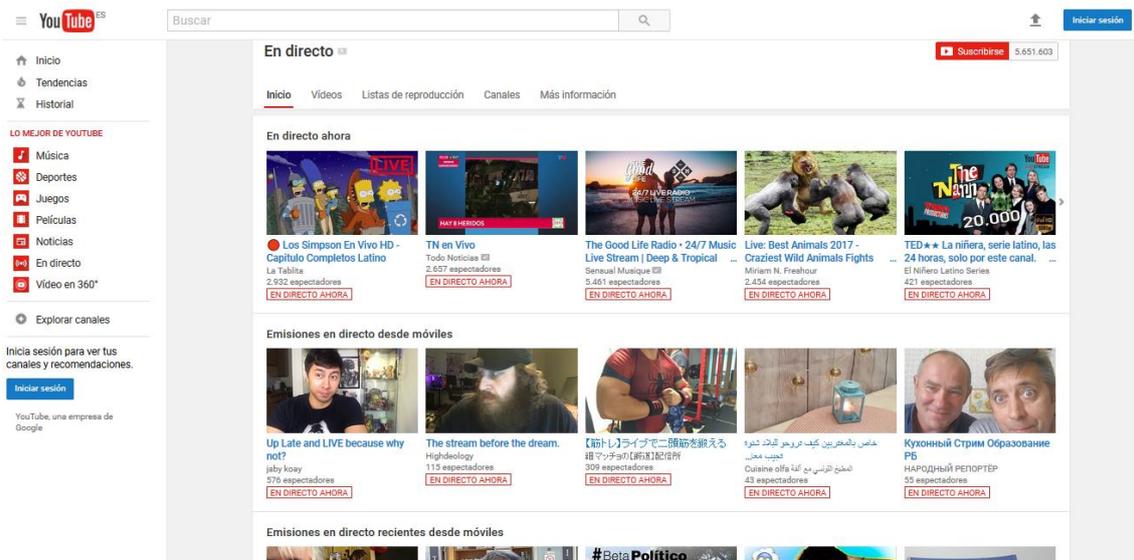


Ilustración 1: Youtube en directo

Periscope

La versión web de la aplicación móvil Periscope permite buscar por una temática, pero no se puede seleccionar la ubicación, ni saber si hay alguna transmisión en directo cerca de tu ciudad o algún lugar de interés. Al igual que Youtube, se echa en falta el poder buscar por ubicación.



Ilustración 2: Periscope versión web

Facebook

Facebook live tiene una interfaz amigable y simple, al seleccionar un marcador se accede al vídeo. Tiene una lista de las emisiones en directo, el único problema es la falta de un buscador por temas ya que para ver de que trata el vídeo se debe acceder a cada marcador o utilizar la ampliación del mapa.

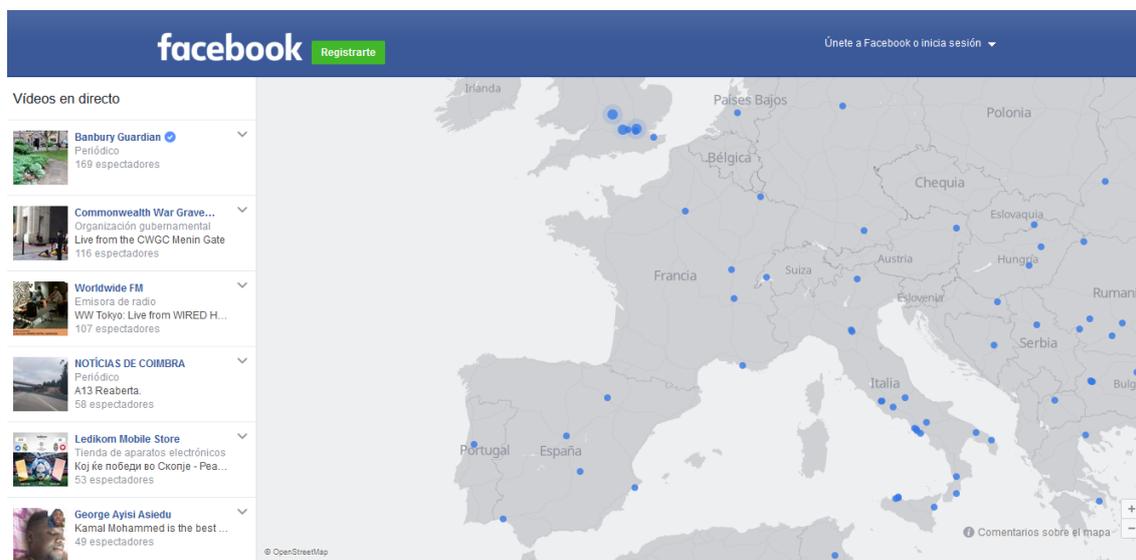


Ilustración 3: Facebook live

A grandes rasgos las funcionalidades básicas que debe cumplir el proyecto son las siguientes:

La aplicación móvil

Para que un usuario pueda acceder a la aplicación móvil y así poder ejecutar sus funcionalidades, debe estar registrado previamente, se pretende gestionar el registro de usuario.

Cuando un usuario está registrado ya puede acceder a la aplicación, en el momento que un usuario quiera salir de la aplicación el usuario debe pulsar la pestaña en cuestión.

Antes de que un usuario pueda enviar en tiempo real el vídeo se debe realizar una configuración para iniciar la emisión, para ello existe un formulario con los campos requeridos de información y geolocalización.

El usuario emisor dispone de los controles o botones necesarios para iniciar o detener la emisión en directo, estos controles permiten enviar en tiempo real el vídeo a los usuarios de la página web.

La página web

En la página principal debe aparecer el mapa con los diferentes marcadores donde se puede realizar una búsqueda por nombre del vídeo, autor o temática. Se muestra una lista dinámica en la cual aparecen los nombres de las emisiones y se actualiza según las conexiones existentes.

Mediante la geolocalización del usuario que está realizando la emisión en directo desde la aplicación móvil, se agrega en el mapa un marcador en la posición obtenida, al salir el usuario de la aplicación móvil el marcador se elimina del mapa.

Al pulsar sobre un marcador se debe redirigir a una nueva ventana que mostrará el vídeo deseado.

3. Metodología de desarrollo

Para la realización de este trabajo se ha optado por usar metodologías ágiles ya que presentan algunas ventajas frente a las metodologías tradicionales y el desarrollo es iterativo e incremental. En la realización de este tipo de desarrollo el trabajo se divide en ítems, estos ítems se agrupan en *sprints* o iteraciones.

Las ventajas de las metodologías ágiles respecto a las metodologías tradicionales son las siguientes:

- La entrega del software es temprana y continua ya que la prioridad es satisfacer al cliente.
- Los cambios de requisitos son bien aceptados, la especificación de requisitos se implementa a medida que se necesita un ítem, es decir, se realiza la especificación de requisitos en cada ítem antes de pasarlo a una iteración para ser desarrollado.
- En cada entrega el software es funcional, se realizan entregas en periodos cortos de tiempo aumentando la funcionalidad.
- Software más mantenible.

Los ítems anteriormente mencionados se han representado mediante las historias de usuario.

Historias de usuario

Las historias de usuario son la representación de los requisitos escritos en pocas frases utilizando el lenguaje natural. Son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentación permitiendo responder de forma rápida a los requisitos cambiantes.

Las historias de usuario deben ser:

- Independientes unas de otras.
- Negociables, la discusión entre los desarrolladores y los usuarios debe permitir esclarecer el alcance y éste debe declararse mediante las pruebas de aceptación.
- Estimables, estimar el tiempo que se tardará en completar la funcionalidad.
- Pequeñas, ya que las historias muy largas son difíciles de estimar.
- Verificables, puesto que cubren los requisitos de los usuarios, son verificables.



Para realizar las historias de usuario se utiliza el siguiente esquema:

Como: [usuario/rol]

Quiero: [funcionalidad]

Para: [objetivo/beneficio]

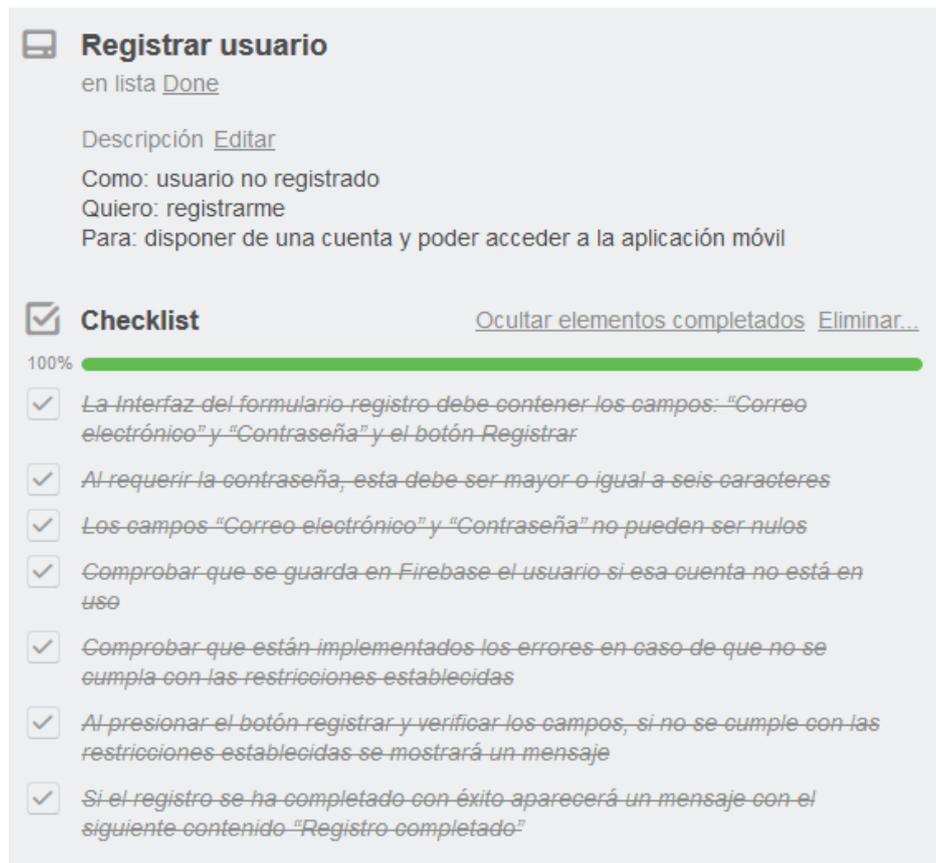


Ilustración 4: Historia de usuario

Pruebas de aceptación

Las pruebas de aceptación consisten en verificar que el sistema desarrollado cumple con las especificaciones y que logra realizar las funciones deseadas por el usuario, por lo tanto, se pueden utilizar para la especificación de requisitos.

Si al esquema del apartado anterior le añadimos "Comprobar que [se cumple la funcionalidad de usuario]" tenemos las pruebas de aceptación.

En las pruebas de aceptación (PAs) no es obligatorio utilizar "Comprobar que", en los apartados donde se especifican las PAs en muchos casos se omite para que no sea repetitivo de escribir y leer.

Kanban

El sistema Kanban virtual nos permite visualizar el estado de las tareas mediante la utilización de una tabla donde cada columna representan las distintas etapas por las que pasa una tarea, se utilizan tarjetas para representar los elementos a realizar (cada tarjeta representa una historia de usuario).

El número de columnas de la tabla depende del proyecto a realizar y de la especificación de cada una de las actividades, en este caso se ha optado por las siguientes columnas:

Backlog: el *backlog* es como el almacén donde están todas las tareas, no es necesario que estén todas especificadas, las de menor prioridad de desarrollo con el nombre es suficiente.

To Do: en esta etapa se encuentra las tareas seleccionadas por prioridad que se desarrollaran en la iteración, aquí se terminan de especificar los requisitos y descripciones.

Doing: en esta columna aparecen las tarjetas que se están implementando, independiente de la fase de desarrollo que se encuentren, también aparece el nombre del desarrollador que está implementando la tarjeta.

Done: al finalizar el desarrollo de una tarjeta se desplaza a esta columna, la cual indica que ha finalizado el desarrollo.

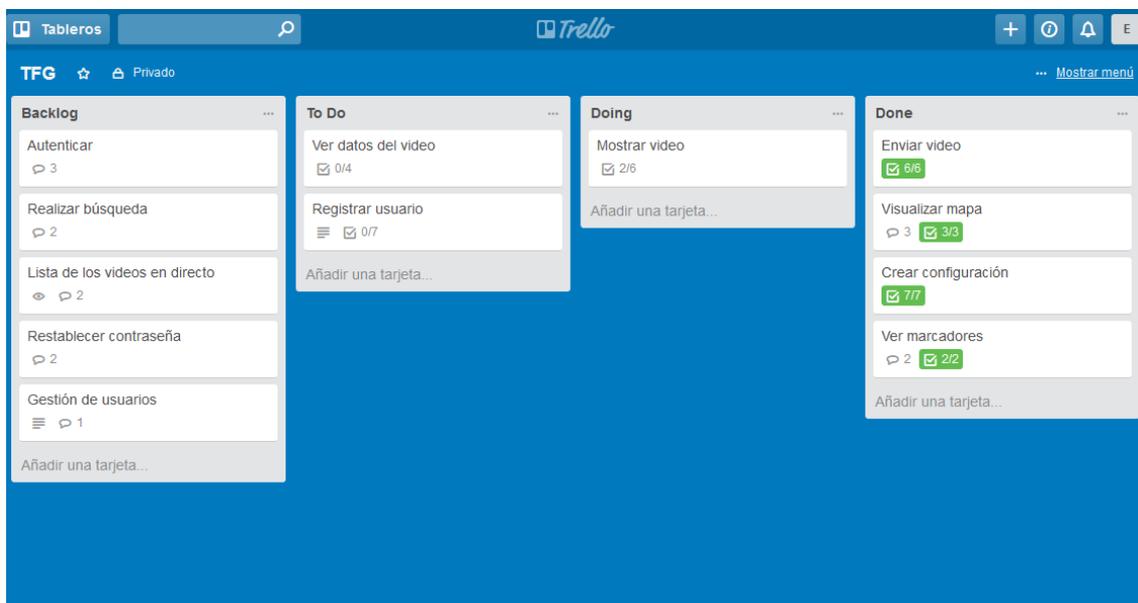


Ilustración 5: Kanban

4. Diseño

Tecnologías utilizadas

A continuación, se enumeran las tecnologías utilizadas para el diseño de este proyecto:

HTML

HyperText Markup Language es un lenguaje de marcas de hipertextos que se utiliza para crear y representar visualmente una página web, es un estándar a cargo de la *World Wide Web Consortium (W3C)*.

CSS

Hojas de estilo en cascada, es un lenguaje de diseño gráfico que describe el estilo de un documento estructurado escrito en HTML, describe cómo deben mostrarse los elementos HTML.

Bootstrap

Bootstrap es un framework para el diseño de aplicaciones y páginas web, el cual contiene plantillas, menús de navegación, botones y otros elementos cuyo diseño está basado en HTML, CSS o JavaScript. Facilita el desarrollo de una interfaz puesto que el diseño es adaptable (*responsive*) y se ajusta a la pantalla de cualquier dispositivo.

Se ha optado por utilizar la versión cuatro de Bootstrap ya que permite tener una interfaz usable realizando pequeñas modificaciones, sin la preocupación de si se adaptará a los diferentes dispositivos que existen.

TypeScript

TypeScript es un lenguaje de programación de código abierto desarrollado por Microsoft. Extiende la sintaxis de JavaScript añadiendo el tipado estático y los objetos basados en clases, de esta forma se programa mediante el paradigma de la programación orienta a objetos (POO).

Al extender de JavaScript permite que el código escrito en JavaScript funcione haciendo pequeñas modificaciones en el código. Los ficheros TypeScript se compilan en ficheros de lenguaje JavaScript original, permitiendo la compatibilidad con todos los navegadores, servidores y sistemas operativos.

Angular

Angular es un framework para desarrollar aplicaciones web mantenido por Google, que utiliza el lenguaje TypeScript y es multiplataforma. Respecto al framework anterior (AngularJS), Angular ha mejorado en cuanto a la escalabilidad, la velocidad, el rendimiento, la mantenibilidad y la productividad.

Para el desarrollo de la parte web del proyecto se ha utilizado la versión cuatro de Angular, se actualizó de la versión dos a la cuatro en el momento que se estaba desarrollando este proyecto. Cabe mencionar la utilización de la herramienta Angular CLI (*Command Line Interface*) que ha facilitado la creación y desarrollo del proyecto.

Ionic

Para la parte de la aplicación móvil se ha optado por el desarrollo de una aplicación híbrida. Una aplicación híbrida es aquella que permite desarrollar aplicaciones para móviles o tabletas usando tecnologías web como la ya mencionadas HTML, CSS, Javascript o TypeScript. En el momento de instalar la aplicación el usuario final no nota la diferencia respecto a las aplicaciones nativas. Las aplicaciones híbridas se ejecutan en un WebView, que es una especie de navegador web interno del móvil.

Ionic versión tres es un framework para el desarrollo de aplicaciones híbridas, el cual utiliza Angular y junto con Apache Cordova facilita el desarrollo de las aplicaciones. Cabe destacar la utilización de la herramienta Ionic CLI que ha facilitado el inicio del proyecto y la creación del esqueleto de la aplicación.

Node.js

Necesitamos instalar Node.js para poder instalar librerías, dependencias y utilidades en Angular CLI e Ionic CLI mediante el gestor de paquetes npm.

WebRTC

WebRTC (*Web Real-Time Communication*) es una tecnología de código abierto que permite a aplicaciones y sitios web realizar llamadas de voz, chat de vídeo o compartir archivos entre navegadores en tiempo real sin la necesidad de instalar complementos (*plug-ins*) o cualquier otro software. WebRTC consta de varias librerías y protocolos que trabajan juntos para lograr este objetivo.

La necesidad de utilizar WebRTC surge por:

- Los programas o servicios existentes web ya utilizan RTC, pero necesitan descargar aplicaciones nativas o complementos. Skype, Hangouts de Google (utiliza el complemento de Google Talk).
- La instalación y actualización de complementos puede ser propensa a errores.
- Los complementos pueden llegar a ser difíciles de depurar, solucionar problemas, probar y mantener. La integración de estos con otras tecnologías existentes puede ser compleja y costosa.



Las principales interfaces de programación de aplicaciones (APIs) de WebRTC son:

- `MediaStream (getUserMedia)`, permite acceder a los flujos de datos de audio y vídeo.
- `RTCPeerConnection`, permite realizar conexiones de audio/vídeo.
- `RTCDataChannel`, posibilita a los navegadores compartir otros tipos de datos como puedan ser datos de juegos en tiempo real, transferencia de archivos, etc.

Los navegadores web que actualmente soportan WebRTC son los que aparecen en verde en la imagen inferior.

IE	Edge *	Firefox	Chrome	Safari	Opera
			49		
		52	58		
	14	53	59		45
11	15	54	60	10.1	46

Ilustración 6: Navegadores web

Los navegadores web disponibles para móvil:

iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
9.3		4.4	
10.2		4.4.4	
10.3	all	56	59

Ilustración 7: Navegadores web para móvil

Crosswalk

Puesto que WebRTC consta de varios componentes, en algunas versiones antiguas de Android esta aplicación no podía funcionar ya que estos componentes no están soportados.

Se ha solucionado el problema instalando el complemento de Cordova Crosswalk puesto que instala un navegador Android (WebView) actualizado con la última versión de Google Chromium.

Firebase

Firebase es una plataforma de trabajo de Google gratuita que permite cien conexiones simultáneas, está orientada al desarrollo móvil. Sirve para que los desarrolladores no tengan que preocuparse en desarrollar y mantener la parte del servidor. Algunas de los servicios que ofrece son:

- Base de datos en tiempo real.
- Sistema de autenticación.
- Hosting.
- Almacenamiento de archivos del usuario.
- Notificaciones a los usuarios.
- Estadísticas sobre el rendimiento de la aplicación.



Arquitectura de WebRTC

WebRTC utiliza una red *peer-to-peer* o red de pares (P2P), es una red de dispositivos en la que todos actúan simultáneamente como clientes y servidores respecto a los demás nodos de la red. Esto significa que los dispositivos intercambian información entre ellos sin la necesidad de acceder a un servidor, obteniendo transferencias más rápidas que en un sistema centralizado.

La conexión entre dos dispositivos se realiza mediante la API `RTCPeerConnection` que se encarga de la comunicación estable y eficiente de los datos, pero la señalización no forma parte de `RTCPeerConnection` y necesita un mecanismo de señalización en el que se intercambia información como:

- Mensajes de control de sesión: para inicializar o cerrar la comunicación, diagnosticar errores en los mensajes.
- Configuración de la red: obtener dirección IP y el puerto del dispositivo.
- Multimedia: identificar códecs y resolución soportada por emisor y receptor.

El intercambio de información de configuración multimedia se realiza intercambiando una oferta y una respuesta utilizando el Protocolo de descripción de sesión (SDP).

La traducción de direcciones de red (NAT) es un mecanismo que permite que múltiples dispositivos compartan una sola dirección IP. NAT traduce la dirección IP interna del cliente a una dirección externa, para los usuarios externos, el tráfico que entra a la red y sale de ella tiene la misma dirección IP.

Los diferentes dispositivos necesitan saber su IP y puerto externo para conectarse entre ellos, como los dispositivos están detrás de un NAT cada dispositivo tiene que acceder a un servidor STUN cuyo cometido es informales de su IP y puerto con el que sale al exterior (dirección IP externa). Para un buen funcionamiento `RTCPeerConnection` necesita tener configurados servidores STUN/TURN (al menos deben existir varios servidores STUN).

El servidor TURN además de hacer la misma función que un servidor STUN estándar, se encarga se reenviar el flujo de información si la conexión entre dos dispositivos (P2P) falla.

El establecimiento interactivo de la conectividad (ICE) es una técnica para encontrar el modo que dos dispositivos puedan conectarse entre ellos tan directamente como sea posible en la red *peer-to-peer*, hace uso de los servidores STUN y TURN.

ICE intenta conectar los dispositivos con la menor latencia posible, mediante el protocolo de transporte UDP y los servidores STUN. Si UDP falla, ICE intenta conectar con el protocolo TCP: primero con HTTP, después con HTTPS. Si sigue sin funcionar utilizará un servidor TURN.

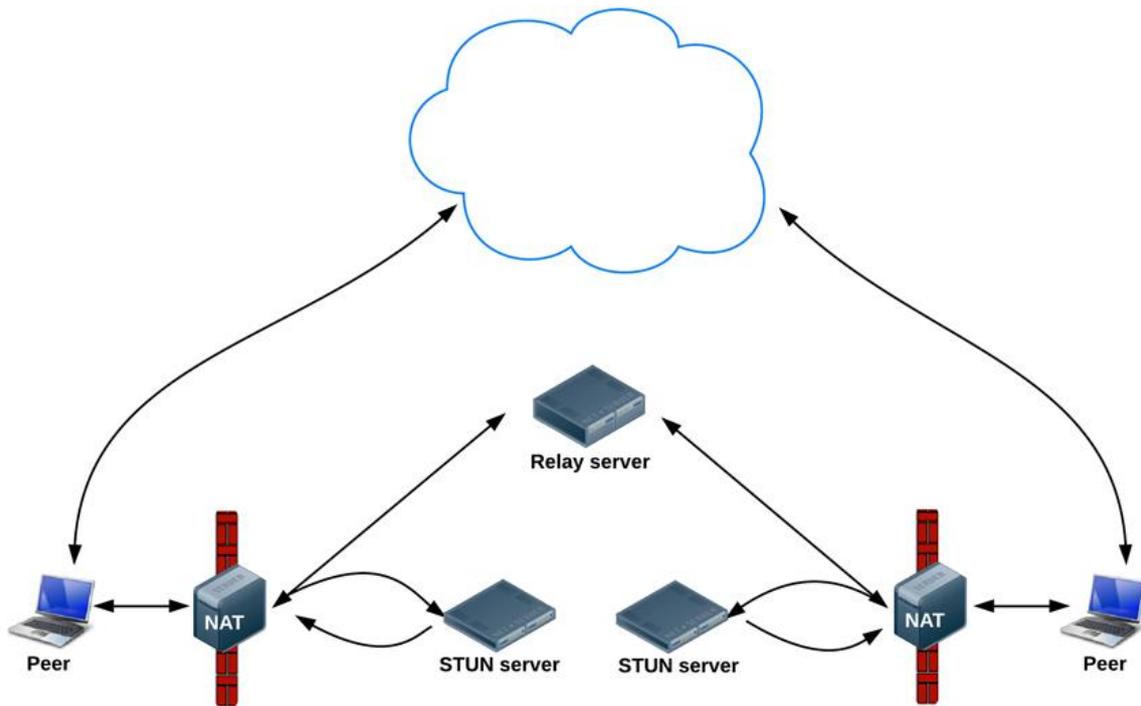


Ilustración 8: Encontrar candidatos en la conexión

Servidor de señalización

Para que los dispositivos puedan intercambiar los mensajes se ha creado el siguiente procedimiento:

Los mensajes SDP oferta, SDP respuesta e ICE se guardan en la base de datos en tiempo real de Firebase codificados en formato JSON, se crea una lista con un identificador por cada usuario que acceder a visualizar el vídeo (usuarios receptores), cuando los mensajes son leídos se borran.

```

webrtc-ccdfe
├── chats
│   ├── 130087305
│   └── 329341297
│       ├── -Kq9NLu661jEELkMzd0R
│       │   ├── id: "899319085"
│       │   └── message: "{\"sdp\":{\"type\":\"offer\",\"sdp\":\"v=0\r\nno-- 487519."
│       ├── -Kq9NLuv3e5tMJUy8b6E
│       │   ├── id: "899319085"
│       │   └── message: "{\"ice\":{\"candidate\":{\"candidate:3805966793 1 udp."
│       ├── -Kq9NLuyc1E4tN656M6P + x
│       │   ├── id: "899319085"
│       │   └── message: "{\"ice\":{\"candidate\":{\"candidate:2999745851 1 udp."
│       ├── -Kq9NLv-IT6AHto8CVdc
│       │   ├── id: "899319085"
│       │   └── message: "{\"ice\":{\"candidate\":{\"candidate:1645705720 1 udp."
│       ├── -Kq9NLv0cO8g_wUDizxq
│       │   ├── id: "899319085"
│       │   └── message: "{\"ice\":{\"candidate\":{\"candidate:1840965416 1 udp."
│       └── -Kq9NLv278_hjBhNyqJ

```

Ilustración 9: Mensajes en la base de datos de Firebase



En la imagen inferior se describe el proceso completo de señalización entre dos candidatos:

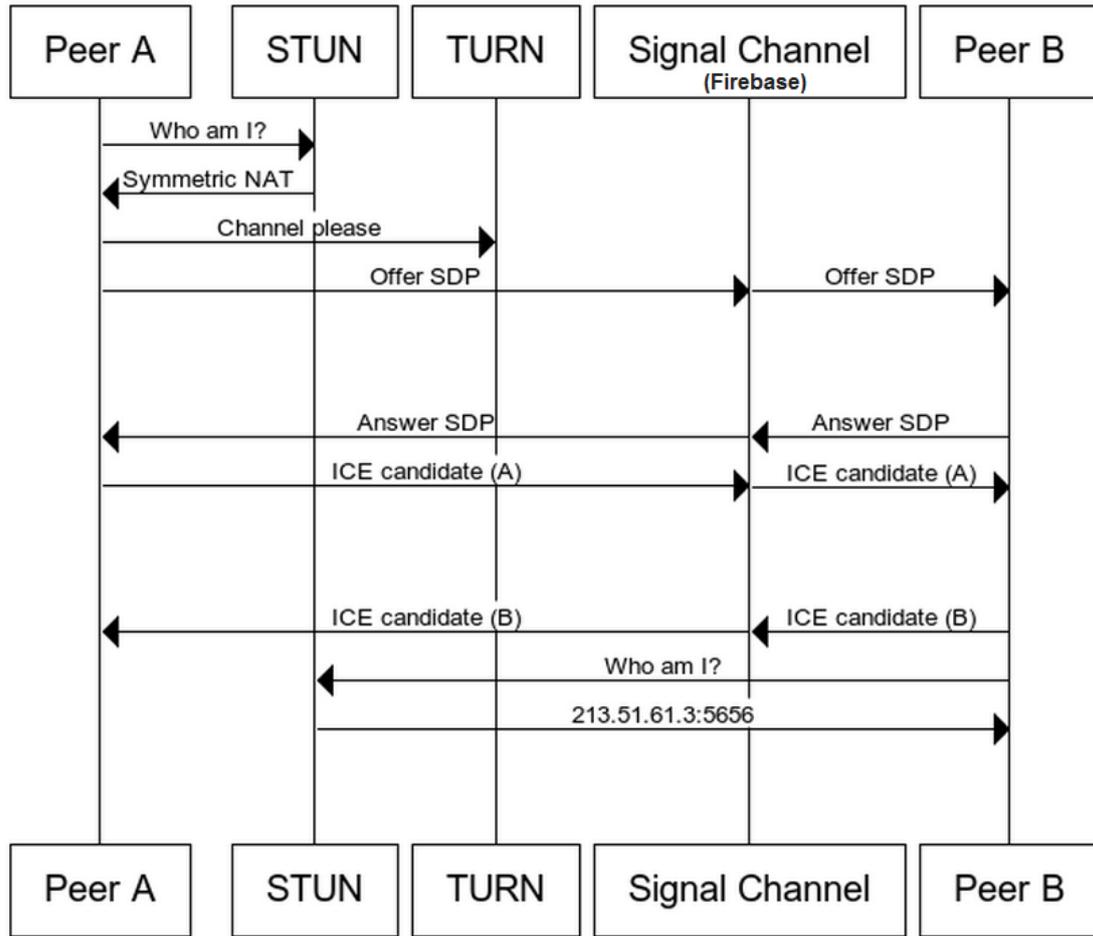


Ilustración 10: Conexión entre dos candidatos

Arquitectura de Angular

La arquitectura de Angular se puede dividir en ocho bloques:

- Módulo
- Componente
- Plantilla
- Metadatos
- *Data Binding*
- Directiva
- Servicio
- Inyección de dependencias

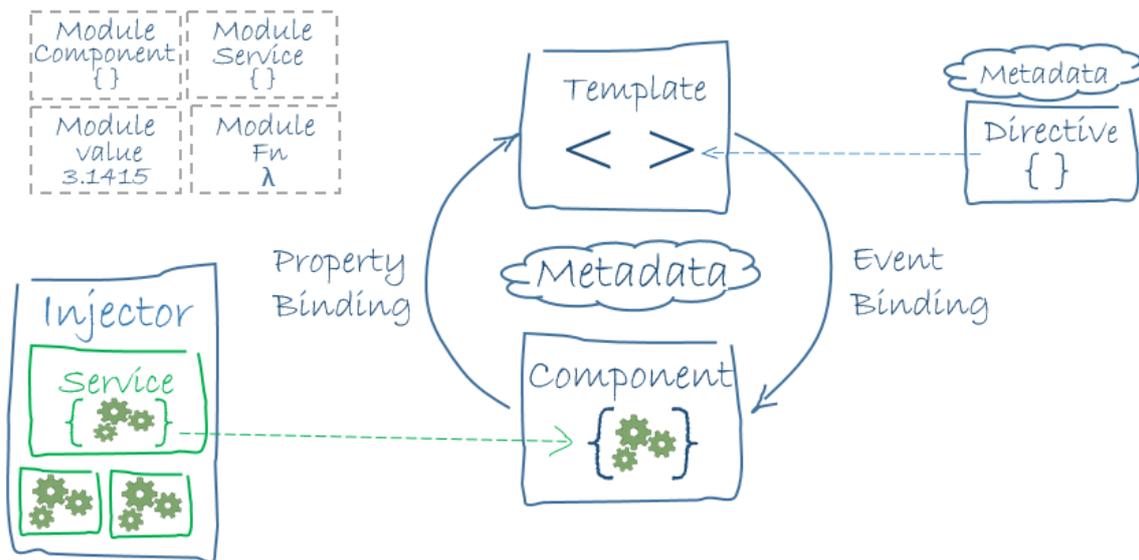


Ilustración 11: Arquitectura de Angular

Módulo

Un módulo es un bloque de código que se utiliza para una funcionalidad específica en una aplicación, se declara mediante el decorador “@NgModules”. Cada aplicación de Angular tiene al menos un módulo, el módulo raíz, por convención es llamado “AppModule” y sirve para cargar la aplicación e indicar todas sus dependencias.

Componente

Un componente controla los elementos que se muestran en la pantalla. Por ejemplo, la barra de navegación, el mapa junto con la lista de los vídeos y el pie de página son componentes. Los componentes implementan métodos y propiedades disponible en su plantilla.

Plantilla

Las plantillas son código HTML con expresiones de Angular e indican cómo se muestra el componente.

Metadatos

Los metadatos son una forma de procesar las clases, mediante el patrón decorador se transforma de una clase a un componente.

Data Binding

Data binding consiste en enlazar la información con la forma de visualización en la plantilla HTML, según lo que se pretenda hacer hay cuatro formas en las que se produce el *data binding*:

- Interpolación, enlaza valores desde el componente hacia el Modelo de Objetos del Documento (DOM).
- *Property binding*, enlaza propiedades del componente hacia el DOM.
- *Event binding*, enlaza eventos desde el DOM hacia el componente.
- *Two-way data binding*, enlaza una propiedad y un evento al mismo tiempo.

Directiva

Las plantillas de Angular son dinámicas, el DOM cambia según el contenido de las directivas. Una directiva es una clase que utiliza el decorador “@Directive”, y sirve para unir la clase con los metadatos, existen directivas estructurales que modifican el diseño y directivas de atributo que alteran la apariencia o el comportamiento de un elemento.

Servicio

En Angular los servicios son clases con cualquier valor, función o característica que necesite la aplicación, estos servicios son consumidos por los componentes.

La inyección de dependencias

La inyección de dependencias es una forma de proporcionar una nueva instancia de una clase con las dependencias que requiere, se realiza en el constructor de una clase y la mayoría de dependencias son servicios que utiliza el componente.

Historias de usuario

Se han implementado las siguientes historias de usuario y pruebas de aceptación ordenadas por orden de prioridad:

Enviar vídeo

Esta funcionalidad permite al usuario enviar en tiempo real el vídeo a los otros usuarios que accedan a la emisión, el usuario emisor dispone de los controles o botones necesarios para iniciar o detener la retransmisión.

Enviar vídeo
Como: usuario de la aplicación móvil Quiero: enviar el vídeo en tiempo real Para: que me puedan ver los usuarios que accedan a la página web

Las pruebas de aceptación (PAs) realizadas son:

- Comprobar que al pulsar el botón “Iniciar” se muestra el siguiente mensaje: “¿Esta seguro que desea iniciar la emisión?” y las opciones de “Volver” y “Aceptar” funcionan adecuadamente.
- Comprobar que al pulsar el botón de “Iniciar” se muestra la imagen de la cámara en directo.
- Al pulsar el botón “Parar” debe tener la emisión en directo.
- Comprobar que la primera vez que se accede a la pantalla “Vídeo WebRTC” aparece el siguiente mensaje “Debe ir a la configuración antes de iniciar la emisión”. Al acceder por segunda vez o volver a dicha pantalla no aparece el mensaje.
- Comprobar que se obtiene la geolocalización del GPS.

Las tareas realizadas para representar esta historia de usuario son:

- Crear la interfaz.
- Crear la funcionalidad.
- Capturar vídeo mediante WebRTC (Crosswalk WebView, plugin Apache Cordova).
- Crear el modelo de datos para almacenar los datos del emisor.
- Guardar en Firebase.
- Test.

Visualizar mapa

En la página principal aparece el mapa con los diferentes controles de aumentar/disminuir el Street View, esta funcionalidad se ha creado a partir de la siguiente historia de usuario.

Visualizar mapa
<p>Como: usuario de la página web Quiero: que al acceder a la página principal aparezca un mapa con los correspondientes controles de aumentar y disminuir el mapa Para: seleccionar los marcadores de diferentes lugares</p>

Las pruebas de aceptación (PAs) realizadas:

- Al entrar a la página principal se muestra el mapa para seleccionar los distintos puntos de interés.
- Comprobar que la funcionalidad de aumentar y disminuir en el mapa funciona correctamente.

Se han realizado las siguientes tareas:

- Crear la interfaz.
- Crear el mapa con Google Maps y propiedades (Angular Google Maps, paquete de npm).

Crear configuración

Antes de que un usuario pueda enviar vídeo en tiempo real se debe realizar una configuración para hacer la transmisión en directo, para ello se ha creado un formulario con los siguientes campos: “Nombre de usuario”, “Título del vídeo”, “Descripción de vídeo”, “Mostrar nombre” y “Tipo de vídeo”. La información es almacenada en Firebase junto con la captura de las coordenadas del GPS.

Crear configuración
<p>Como: usuario de la aplicación móvil Quiero: personalizar la información a enviar Para: que el vídeo pueda ser visto por más gente</p>

Las respectivas PAs implementadas:

- Los campos del formulario “Nombre de usuario” y “Título del vídeo” no pueden ser nulos.
- Comprobar que en el desplegable “Tipo de vídeo” se puedan seleccionar las opciones mediante el botón “OK” o “Cancelar”.
- La descripción no tiene valor por defecto.
- Al salir del campo “Nombre de usuario” o “Título del vídeo”, sino se ha introducido datos y se pulsa “Guardar”, debe salir un mensaje explicativo.

- Comprobar que al guardar y volver a acceder de nuevo a la pestaña de configuración, los datos introducidos se han guardado correctamente.
- Comprobar que se obtiene la geolocalización del usuario.

Se han implementado las siguientes funciones:

- Crear la interfaz del formulario.
- Crear la funcionalidad del formulario.
- Obtener las coordenadas mediante el complemento de geolocalización (*Geolocation*, complemento de Apache Cordova).
- Guardar datos de la configuración en Firebase.
- Test.

Ver marcadores

Mediante la geolocalización almacenada de un usuario desde la aplicación móvil se implementa un marcador en la posición que indica la geolocalización obtenida, al salir el usuario de la aplicación móvil el marcador es eliminado del mapa.

Ver marcadores
Como: usuario de la página web Quiero: poder ver los diferentes marcadores Para: poder seleccionar los que estén cerca de mi ciudad

Las PAs son las siguientes:

- Comprobar que se muestran los marcadores de las distintas coordenadas guardadas que existen en el momento de acceso a la página web.
- Al iniciar una emisión un usuario, se debe insertar un nuevo marcador en el mapa con las coordenadas de dicho usuario.
- Al salir de una emisión un usuario emisor el marcador es eliminado.

Las tareas realizadas para representar esta historia de usuario son:

- Implementar marcadores.
- Implementar la funcionalidad mostrar/eliminar marcador.
- Crear información.
- Test.

Mostrar vídeo

Para poder acceder a un vídeo se tiene que pulsar sobre un marcador, para ello se ha creado un enlace para cada marcador.



Mostrar vídeo

Como: usuario de la página web

Quiero: acceder a un vídeo

Para: poder ver los eventos que se están realizando en un momento dado

Con las siguientes PAs:

- Comprobar que al pulsar sobre un marcador, este redirige hacia una nueva página con el vídeo correspondiente y se visualiza.
- Los vídeos se corresponden con la emisión del usuario, es decir, al pulsar sobre los marcadores que aparecen en el mapa todos tienen un vídeo distinto.
- Comprobar que no aparecen marcadores sin vídeo.

Las tareas realizadas son:

- Crear el diseño de la nueva página.
- Desarrollar la funcionalidad para ver el vídeo.
- Implementar la funcionalidad del enlace.
- Test.

Ver datos del vídeo

El usuario puede tener información sobre el vídeo que está viendo, como la descripción del vídeo o el nombre del usuario que está haciendo esa transmisión en directo.

Ver datos del vídeo

Como: usuario de la página web

Quiero: ver información sobre un vídeo, nombre del usuario, descripción y la temática de la transmisión en directo

Para: poder seleccionar los vídeos más interesantes

PAs implementadas:

- Comprobar que, al pulsar sobre un vídeo se redirige hacia una nueva página con el vídeo correspondiente y aparece el nombre del usuario y la correspondiente descripción si se ha insertado.
- El campo “Descripción” no es obligatorio, puede ser nulo.
- El campo “Nombre del usuario” es un campo obligatorio, pero puede que no se muestre.

Se han realizado las siguientes tareas:

- Diseño de la interfaz.
- Funcionalidad para ver el vídeo en directo.

- Recuperar la descripción y nombre de usuario.

Registro usuario

Para que un usuario pueda acceder a la aplicación móvil y así poder ejecutar sus funcionalidades debe estar registrado previamente, con esta historia de usuario se gestiona el registro de un usuario.

Registrar usuario
Como: usuario no registrado Quiero: registrarme Para: disponer de una cuenta y poder acceder a la aplicación móvil

Las PAs realizadas:

- La interfaz del formulario registro debe contener los campos: “Correo electrónico” y “Contraseña” y el botón Registrar.
- Al requerir la contraseña, esta debe ser mayor o igual a seis caracteres.
- Los campos “Correo electrónico” y “Contraseña” no pueden ser nulos.
- Comprobar que se guarda en Firebase el usuario, si esa cuenta no está en uso.
- Comprobar que están implementados los errores en caso de que no se cumpla con las restricciones establecidas.
- Al presionar el botón registrar y verificar los campos, si no se cumple con las restricciones establecidas se mostrará un mensaje.
- Si el registro se ha completado con éxito aparecerá un mensaje con el siguiente contenido “Registro completado”.
- Al registrar un usuario se le enviará correo para dar de alta dicho usuario.
-

Tareas realizadas:

- Crear la interfaz del formulario registro con los campos especificados.
- Dar de alta en Firebase en caso de que esa cuenta no esté en uso.
- Implementación de errores al realizar la operación de registrar.
- Test.

Autenticar

Cuando un usuario ya está registrado puede acceder a la aplicación mediante el correo y la contraseña indicada en el registro. En el momento que un usuario quiera salir de la aplicación el usuario debe pulsar la pestaña en cuestión.



Autenticar
Como: usuario registrado Quiero: acceder a la aplicación Para: poder realizar una transmisión en directo de vídeo

Las PAs son las siguientes:

- Comprobar que si no se introducen datos en los campos “Correo del usuario” y “Contraseña” al pulsar “Acceder”, aparece el siguiente mensaje: “Debe introducir todos los campos”.
- Al insertar el correo y contraseña en el registro y pulsar el botón “Acceder”, si el correo introducido es incorrecto se verá el siguiente mensaje “El campo correo no es correcto”.
- Si la contraseña introducida es incorrecta aparecerá un mensaje explicativo.
- Dentro de la aplicación si pulsamos la pestaña “Salir” se mostrará “¿Está seguro de que desea salir?” con las opciones de “Volver” y “Salir”, si se pulsa esta última se cierra la aplicación.

Tareas realizadas:

- Crear la funcionalidad para acceder.
- Implementar errores.
- La funcionalidad salir.
- Test.

Lista de los vídeos en directo

En la página principal se ha implementado una lista dinámica en la cual aparecen el nombre de las emisiones, la lista se actualiza según se conectan y desconectan los usuarios.

Lista de los vídeos en directo
Como: usuario de la página web Quiero: ver información sobre los nuevos vídeos en directo Para: saber cuáles son las emisiones más recientes

Las PAs son las siguientes:

- En el momento que se crea un nuevo vídeo aparecerá el nombre en la lista de la pantalla principal.
- Comprobar que al terminar una emisión desaparece el nombre de la lista en la pantalla principal.

Las tareas realizadas son:

- Diseño de la lista.
- Funcionalidad añadir/eliminar un nombre de vídeo de la lista.

Realizar búsqueda

Permite realizar una búsqueda por nombre del vídeo, con la posibilidad de utilizar un filtro por nombre del autor del vídeo y la temática. Las temáticas pueden ser de tipo: espectáculos, música, deportes, noticias y otros.

Realizar búsqueda
Como: usuario de la página web Quiero: realizar búsqueda según mis intereses Para: ver el contenido disponible

PAs realizadas:

- Al buscar un nombre del vídeo y pulsar el botón “Buscar” aparece el marcador correspondiente.
- Comprobar que al presionar “Filtro de búsqueda” e insertar un nombre de usuario en el campo “Nombre” y presionar el botón “Guardar”, los campos introducidos se quedan guardados para realizar una búsqueda.
- Si se presiona el “Filtro de búsqueda” al insertar una temática desde el desplegable y acto seguido se presiona el botón “Buscar” aparecen los marcadores filtrados por la temática elegida.
- Al pulsar sobre el “Filtro de búsqueda”, el campo “Temática” contiene el valor “Todos” por defecto.

Las tareas implementadas son las siguientes:

- Se ha diseñado el estilo de los elementos que componen la funcionalidad.
- Diseño de la funcionalidad buscar.
- Se ha implementado el filtro de búsqueda.
- Test unitarios.

Restablecer contraseña

En la primera pantalla de la aplicación móvil aparece un enlace que permite recuperar una contraseña de un usuario mediante el correo electrónico.

Restablecer contraseña
Como: administrador Quiero: que los usuarios puedan restaurar la contraseña mediante el envío de un correo Para: seguir disfrutando de la aplicación en caso del olvido de la contraseña



PAs realizadas:

- Al ingresar en la pantalla principal debe aparecer el botón para restablecer la contraseña.
- Si un usuario pulsa el botón “Restablecer contraseña”, se le enviará un correo a dicho usuario en la que aparecerá una plantilla para restablecer la contraseña.

Las tareas realizadas son las siguientes:

- Funcionalidad para restablecer la contraseña.
- Envío del correo al usuario para dicho propósito.

Gestión de usuarios

En la primera pantalla de la aplicación móvil aparece un enlace en el que el usuario que desee tener información o necesite ayuda puede ponerse en contacto con el administrador.

Gestión de usuarios
Como: administrador Quiero: ayudar a los usuarios si tienen alguna duda Para: que puedan disfrutar de la aplicación

PAs realizadas:

- Al ingresar en la pantalla principal debe aparecer una dirección del correo a la cual pueda contactar el usuario.
- Si un usuario envía un correo sobre alguna pregunta referente a la aplicación, el administrador se encargará de gestionar ese correo y enviará lo más pronto posible una respuesta.

Tareas realizadas:

- Mostrar información de contacto mediante el correo.

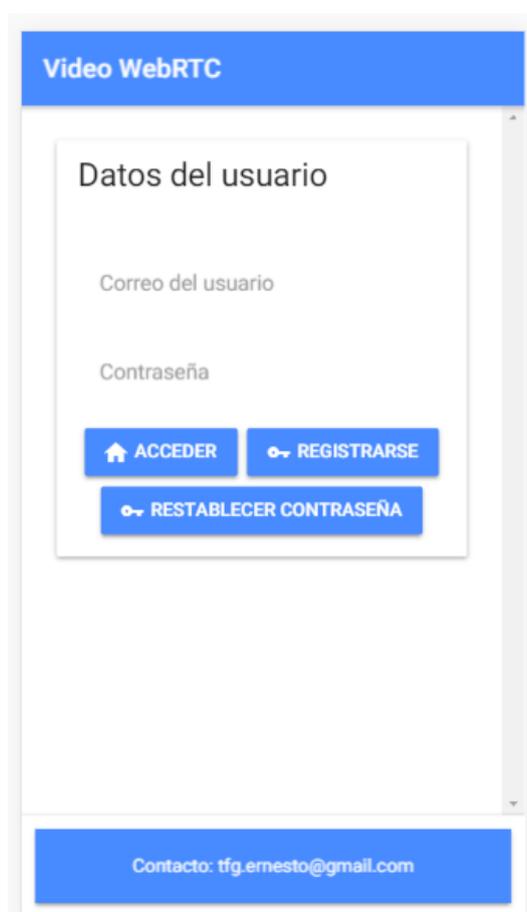
5. Desarrollo

Interfaz de usuario

Registro de usuarios

Como ya se ha comentado en las historias de usuario y las PAs, para la autenticación en la aplicación móvil se ha usado el correo electrónico ya que mucha gente dispone de correo, pero puede que no disponga de redes sociales, por lo tanto, es una forma de que esta aplicación sea usada por más gente.

Cuando una persona accede por primera vez a la aplicación y no tiene usuario debe registrarse introduciendo los campos “Correo del usuario” y “Contraseña”.



The screenshot shows a mobile application interface for user registration. At the top, there is a blue header with the text "Video WebRTC". Below the header, the main content area is titled "Datos del usuario" and contains two input fields: "Correo del usuario" and "Contraseña". Below these fields are three blue buttons: "ACEDER" (with a home icon), "REGISTRARSE" (with a right-pointing arrow icon), and "RESTABLECER CONTRASEÑA" (with a circular arrow icon). At the bottom of the screen, there is a blue footer bar with the text "Contacto: tfg.ernesto@gmail.com".

Ilustración 12: Pantalla de acceso

Al pulsar el botón “Registrarse”, en caso de que falte algún campo o no estén los datos bien introducido, nos aparecen los siguientes errores:

- Si el campo correo no está correctamente introducido, se mostrará el mensaje de la imagen inferior.



Ilustración 13: Mensaje correo incorrecto

- En caso de que el correo ya esté registrado.

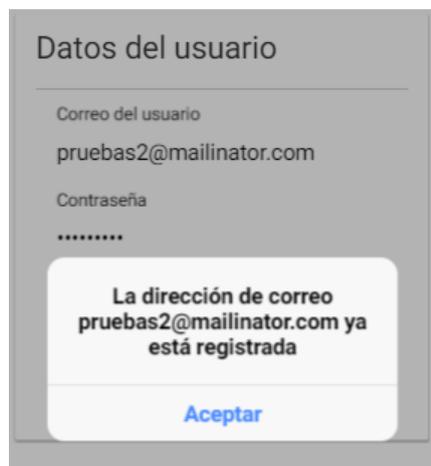


Ilustración 14: Dirección ya registrada

- Si la contraseña no alcanza la longitud mínima (ilustración 15).

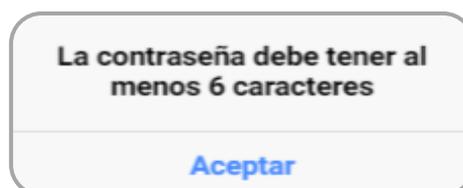


Ilustración 15: Longitud mínima de la contraseña

- Si por el contrario los datos están bien introducidos y pulsamos el botón “Registrar”, se visualizará el mensaje de la ilustración inferior.

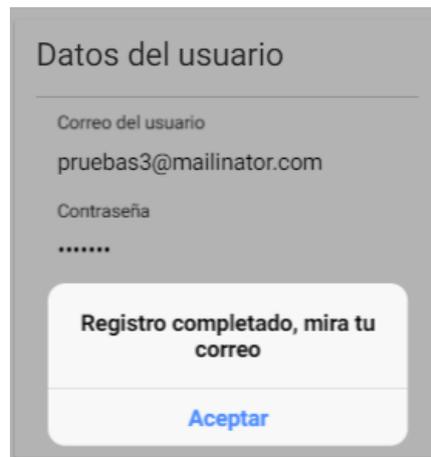


Ilustración 16: Registro completado

Acto seguido en la dirección de correo electrónico introducida nos llegará un correo de verificación. Este correo nos permite saber que el correo introducido por el usuario en el registro realmente pertenece a dicho usuario y así evitar problemas de suplantación de identidad.

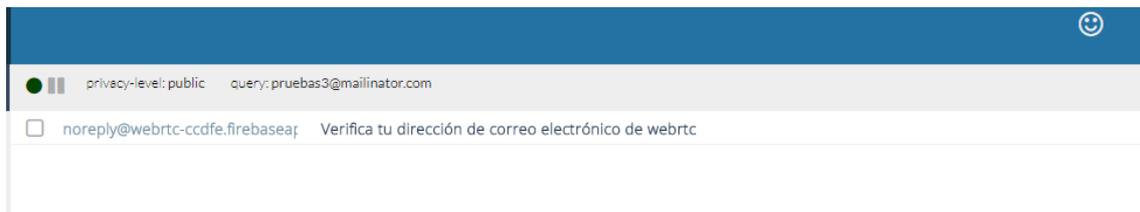


Ilustración 17: Correo electrónico

Al entrar al correo, el contenido será el siguiente:



Ilustración 18: Contenido del correo

Al pulsar sobre el enlace nos indica que ya podemos usar el usuario introducido.

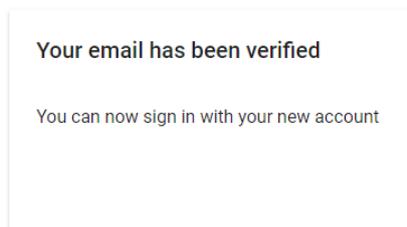


Ilustración 19: Mensaje de verificación

Acceso

Una vez que ya hemos hecho los pasos anteriores, ya podemos acceder. En el caso que no introduzcamos los datos correctamente en el formulario y pulsemos "Acceder", nos aparece además de los errores anteriormente mencionados los siguientes:

- Si nos hemos equivocado al insertar el campo "Correo del usuario" y la dirección no está en uso aparece el mensaje de la ilustración 20.

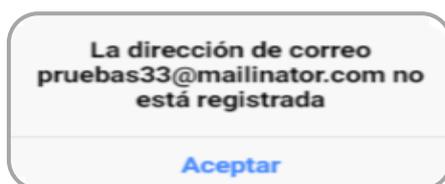


Ilustración 20: Dirección de correo no registrada

- Al acceder el campo "Correo del usuario" es correcto pero la contraseña es incorrecta nos aparece el mensaje de la imagen inferior.

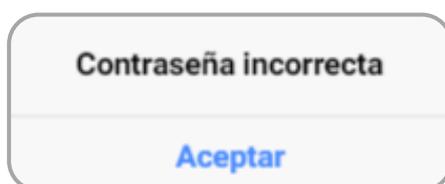
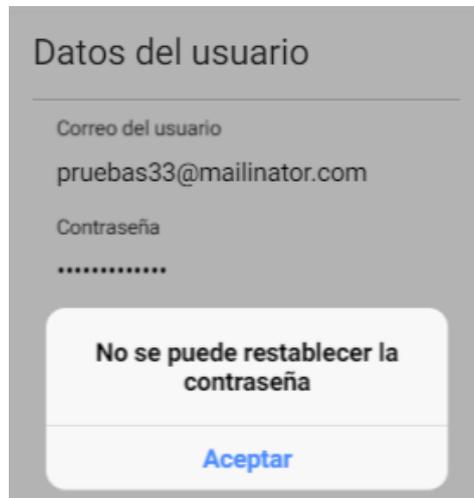


Ilustración 21: Contraseña incorrecta

Restablecer contraseña

Esta funcionalidad permite restablecer una contraseña, es decir un usuario ha olvidado la contraseña y desea introducir una nueva. Los pasos a seguir son los siguientes:

Se introducen los datos de igual forma que el registro o la autenticación, si el correo electrónico introducido no ha sido dado de alta en el sistema con anterioridad, nos muestra el mensaje de la ilustración 22.



Datos del usuario

Correo del usuario
pruebas33@mailinator.com

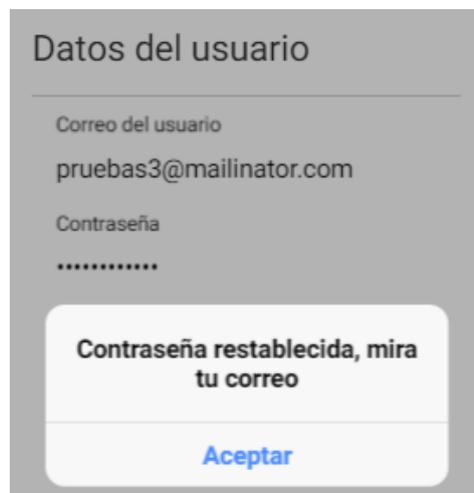
Contraseña
.....

No se puede restablecer la contraseña

Aceptar

Ilustración 22: Error al restablecer la contraseña

En el caso que los datos estén bien introducidos, se muestra el aviso correspondiente y en el correo electrónico tendremos un nuevo correo.



Datos del usuario

Correo del usuario
pruebas3@mailinator.com

Contraseña
.....

Contraseña restablecida, mira tu correo

Aceptar

Ilustración 23: Restablecer contraseña



Ilustración 24: Correo recibido



Ilustración 25: Contenido del correo

Si accedemos al correo y pulsamos sobre el enlace, nos redirige al siguiente formulario, al introducir una nueva contraseña ya podremos usar la cuenta.

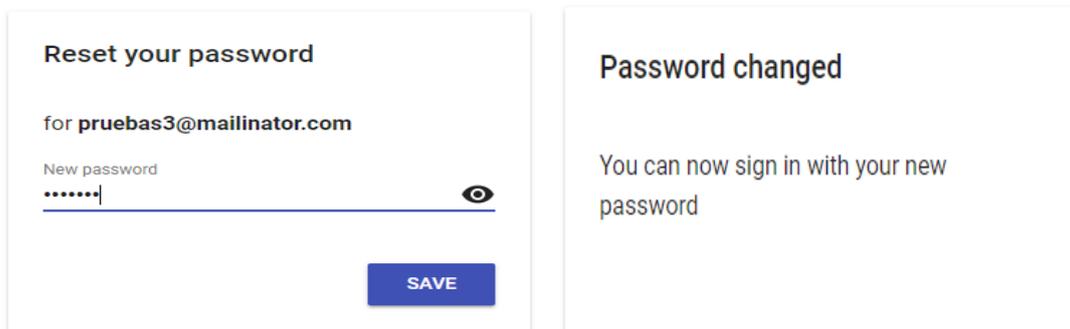


Ilustración 26: Pantalla para restablecer la contraseña

Crear una emisión en directo

Una vez completado el proceso de autenticación ya podemos crear una emisión en directo. En la página “Home”, nos aparece el mensaje “Debe ir a la configuración antes de iniciar la emisión” y los botones para inicializar y parar la transmisión en directo están bloqueados.

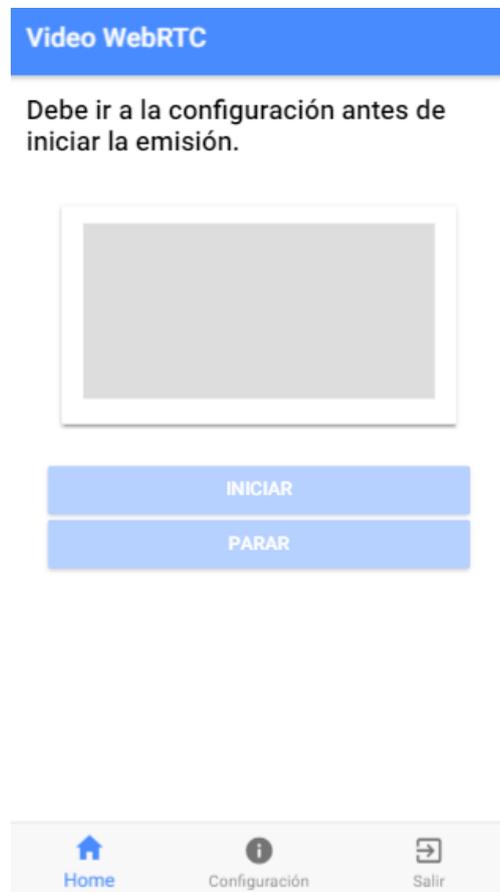


Ilustración 27: Pantalla principal

Si accedemos a la pestaña “Configuración” nos aparecen los campos de la imagen inferior de los cuales “Nombre de usuario” y “Título del vídeo” son obligatorios. A pesar de que el “Nombre de usuario” es obligatorio, el campo “Mostrar nombre” permite a un usuario ocultar el nombre durante la emisión.

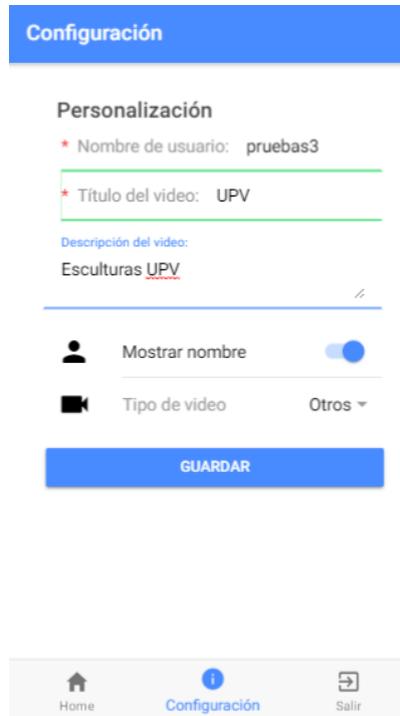


Ilustración 28: Configuración

El siguiente campo “Tipo de vídeo” permite seleccionar una temática entre una serie de temáticas.

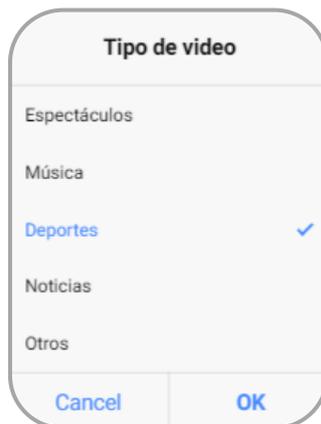


Ilustración 29: Desplegable

Para guardar la configuración realizada solo hace falta pulsar “Guardar” y ya podemos iniciar una transmisión en directo. Dado que ya tenemos una configuración realizada, si accedemos a la pestaña “Home”, al pulsar el botón “Iniciar” se inicia el envío del vídeo en tiempo real.



Ilustración 30: Emisión en directo

La página web

En la página principal de la web aparece el mapa en el que se puede reducir y aumentar el tamaño, buscar con las diferentes opciones de búsqueda o establecer contacto mediante Facebook, Twitter o Google+.

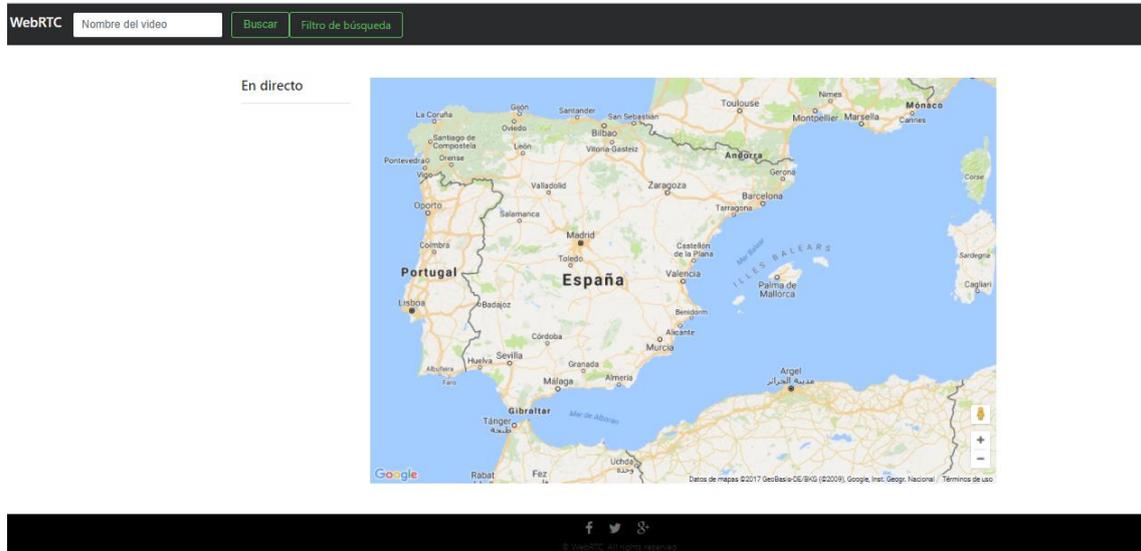


Ilustración 31: Página principal

En el momento que se crea una nueva emisión en directo aparece el nombre de esta en la lista “En directo” y su correspondiente marcador, al terminar desaparecen ambos. Si pasamos el ratón sobre un marcador aparece el nombre del vídeo.



Ilustración 32: Mapa con marcadores

Al pulsar sobre un marcador, vamos a una ventana en la cual podemos ver el vídeo en directo, con el nombre del autor y una descripción.



Ilustración 33: Marcador seleccionado

En la pantalla principal disponemos de la funcionalidad buscar que permite buscar por nombre del vídeo. Al pulsar sobre el botón “Filtro de búsqueda”, podemos aplicar unos filtros de búsqueda sobre los campos “Usuario” y “Temática”.



Ilustración 34: Filtro de búsqueda

EL campo “Temática” contiene las siguientes opciones para elegir:

Temática

Todos
Todos
Espectáculos
Música
Deportes
Noticias
Otros

Ilustración 35: Desplegable temática

6. Evaluación de las prestaciones

Para evaluar las prestaciones se ha utilizado `chrome://webrtc-internals/` cuyo cometido es presentar información sobre la conexión, parámetros y errores de una conexión WebRTC establecida.

La evaluación se ha realizado en una red wifi con un ancho de banda de 50 Megabits por segundo (Mbps) para el emisor y la misma red wifi conectada a una VPN con un ancho de banda de 10 Mbps para el receptor.

En el lado del emisor obtenemos los siguientes valores:

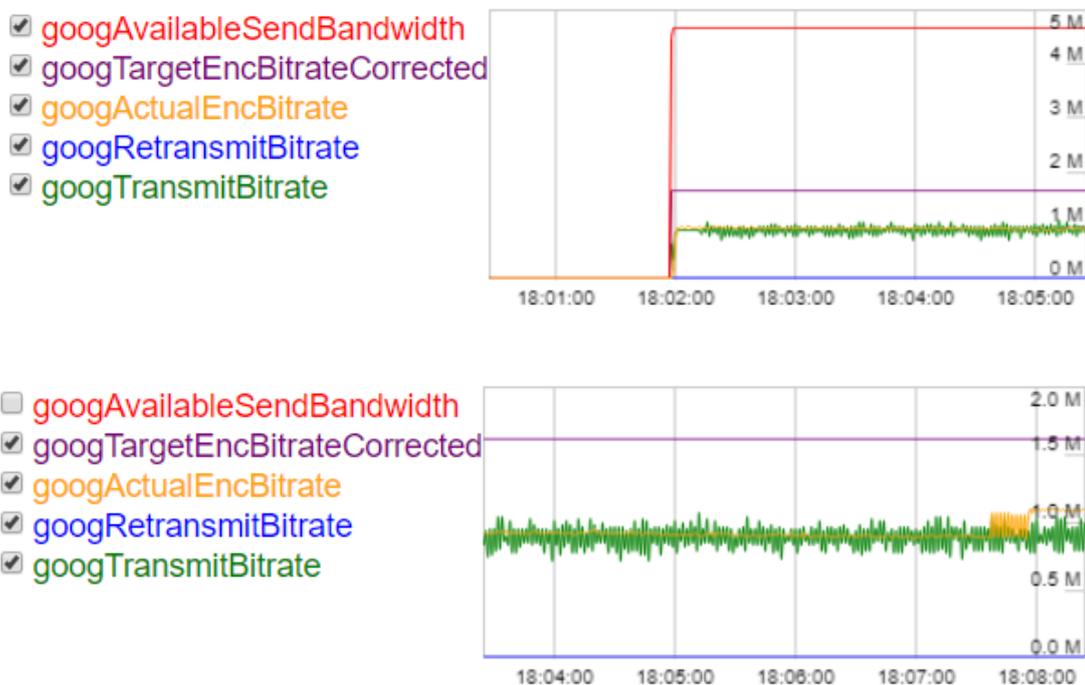


Ilustración 36: Transferencia emisor

`GoogAvailableSendBandwidth` es el ancho de banda disponible para enviar datos, en este caso está cerca de los 5 Mbps.

`GoogTargetEncBitrateCorrected` se refiere a la velocidad del codificador de vídeo en el receptor. El valor de este parámetro aproximadamente es 1,6 Mbps.

`GoogActualEncBitrate` es la tasa de bits que sale del codificador de vídeo, se aprecia en la imagen que el valor llega a un pico de 1Mbps.

GoogTransmitBitrate se refiere a la velocidad de transmisión realmente transmitida. Si es muy diferente de la tasa de bits del codificador, esto puede ser debido a la corrección de errores. Se aprecia en la imagen anterior que el valor está muy relacionado con el valor de GoogActualEncBitrate.

GoogRetransmitBitrate permite medir la tasa de datos (*bitrate*) de una emisión y puede indicar una pérdida de paquetes.

En el receptor:



Ilustración 37: Ancho de banda receptor

El parámetro “GoogAvailableReceiveBandwidth” es el ancho de banda disponible para recibir datos, como se muestra en la imagen anterior, el ancho de banda del receptor es mayor a los datos enviados por el emisor, así pues, no tenemos problemas por insuficiente ancho de banda.

Existen muchos valores, pero vamos a analizar los siguientes:

bytesReceived	411831
mediaType	audio
packetsLost	0
packetsReceived	4396
ssrc	2225303989
transportId	Channel-audio-1
googAccelerateRate	0
googCaptureStartNtpTimeMs	0
googCodecName	opus
googCurrentDelayMs	95
googDecodingCNG	0
googDecodingCTN	8825
googDecodingCTSG	0
googDecodingMuted	28
googDecodingNormal	8789
googDecodingPLC	7
googDecodingPLCCNG	29
googExpandRate	0

Ilustración 38: Parámetros audio

Entre los campos más importantes que podemos ver, campo “mediaType” nos indica que estos parámetros son para el audio y está utilizando el códec opus (campo “googCodecName”) nos aparece un retardo de 95 milisegundos. El campo “bytesReceived” nos informa que se han recibido 411831 bytes, el número de paquetes recibidos (“packetsReceived”) son 4396 paquetes. Con el campo “packetsLost” podemos saber la calidad de la emisión ya que nos indica si se ha perdido algún paquete, en este caso no se ha perdido ninguno.

En la parte del vídeo tenemos la siguiente información:

bytesReceived	17182656
codeImplementationName	libvpx
framesDecoded	2626
mediaType	video
packetsLost	0
packetsReceived	15989
ssrc	2043935570
transportId	Channel-audio-1
googCaptureStartNtpTimeMs	0
googCodecName	VP8
googCurrentDelayMs	51
googDecodeMs	2
googFirsSent	0
googFrameHeightReceived	480
googFrameRateDecoded	20
googFrameRateOutput	20
googFrameRateReceived	20
googFrameWidthReceived	360

Ilustración 39: Parámetros vídeo

Entre los campos más importantes que podemos ver, está el campo “mediaType” nos informa de que los parámetros de la ilustración 39, están relacionados con el vídeo, se utiliza para la emisión el códec VP8 (campo “CodecName”) nos aparece un retardo de 51 milisegundos. El campo “bytesReceived” indica 17182656 bytes recibidos y en “packetsReceived” 15989 paquetes recibidos.

El campo “*googFrameHeightReceived*” nos indica 480, es la altura del último fotograma de vídeo procesado, “*googFrameHeightReceived*” es 360 y es el ancho del último fotograma de vídeo procesado.

Podemos observar el campo “*googFrameRateReceived*” el número total de tramas correctamente recibidas que son 20, el número de tramas decodificadas en “*googFrameRateDecoded*” que son 20 y “*googFrameRateOutput*” representa las tramas de salida que son las decodificadas. De igual modo que en el audio el campo “*packetsLost*” nos indica que no tenemos ningún error.

Audio



Ilustración 40: Grafica audio

Vídeo

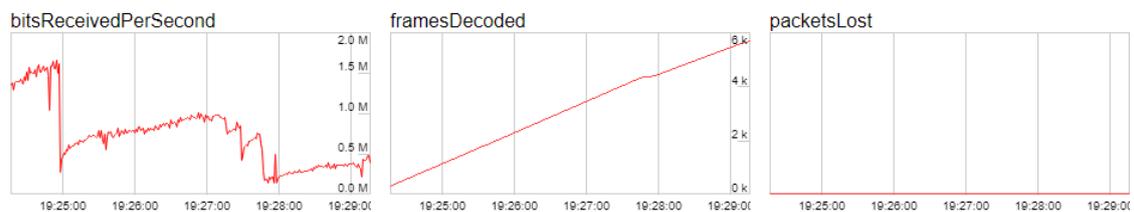


Ilustración 41: Grafica vídeo

Los valores obtenidos son de una conexión con un ancho de banda de 50 Mbps del emisor y 10Mbps del receptor, como hemos mencionado anteriormente tenemos un ancho de banda para establecer una comunicación con garantías y de calidad, aunque la aplicación tiene pequeños retardos se puede establecer una buena comunicación.

En comunicaciones en tiempo real el usuario busca la calidad de la imagen y la rapidez en la transmisión, esto significa, tener un buen ancho de banda para mostrar el vídeo con una calidad aceptable y que el retraso sea mínimo ya que no es agradable ver un vídeo cuyo sonido va a un ritmo diferente del vídeo o el vídeo se bloquea y va a “golpes”.

Para que esta aplicación móvil funcione bien el ancho de banda en el lado emisor debe ser mínimo de 1Mbps y en el receptor similar o un poco más, sobre 1,6 Mbps sería lo adecuado para no producir cuellos de botella, tener un retardo menor y no tener pérdida de paquetes.

En velocidades inferiores se perderá calidad en la visualización y se aumentará el retardo en el vídeo, si bien es cierto que con menos ancho de banda funcionará perdiendo calidad, si el ancho de banda del emisor o receptor está muy limitado la emisión no satisfará a los usuarios y se producirán efectos no deseados.

7. Conclusiones

Después de unos meses de estudio y aprendizaje de las tecnologías no usadas hasta el momento, incluso antes de iniciar el TFG y con el desarrollo del proyecto finalizado podemos llegar a las siguientes conclusiones:

Se ha logrado el objetivo de desarrollar una aplicación para móviles Android la cual permite enviar un vídeo en tiempo real a una página web con una funcionalidad aceptable, pero el vídeo solo está disponible en la página web si se accede desde el navegador Google Chrome, debido a las limitaciones de tiempo no se ha podido corregir este problema (una posible solución en el apartado siguiente).

El resultado de la evaluación muestra como funciona esta aplicación para un ancho de banda en el emisor de 50 Mbps y en el receptor de 10 Mbps, se aprecia un pequeño retardo tanto en el vídeo como en el audio, pero es aceptable y se puede establecer una emisión en directo con buena calidad.

La calidad del vídeo, la pérdida de paquetes y el retardo dependerá del ancho de banda tanto del emisor como del receptor. WebRTC varía la calidad de vídeo y sonido según el ancho de banda disponible, esto se hace en los paquetes SDP ya mencionados en el apartado de la Arquitectura de WebRTC.

A título personal he logrado aprender nuevas tecnologías que seguro que me servirán en el futuro y el hecho de enfrentarme a problemas supone un paso más para crecer como programador.

Ampliaciones futuras

Aunque este TFG es funcional, una posible solución para poder acceder desde todos los navegadores podría ser el uso de `adpater.js`. `Adapter.js` es un complemento que abstrae a la aplicación de cambios en los prefijos o especificaciones de los navegadores de esta forma podría funcionar con cualquier navegador.

Otra posible mejora es la utilización de librerías o APIs ya existentes como `PeerJS` o `SimpleWebRTC`. Estas APIs corrigen algunos problemas y facilitan la creación de aplicaciones sin preocuparse de cosas como el intercambio de mensajes, el servidor de señalización etc.



8. Bibliografía

- [1] **W3C**. *WebRTC 1.0: Real-time Communication Between Browsers* [online]. <https://w3c.github.io/webrtc-pc/archives/20170605/webrtc.html>
- [2] **WebRTC** [online]. <https://webrtc.org/>
- [3] **Sam Dutton**. *Getting Started with WebRTC* [online]. <https://www.html5rocks.com/en/tutorials/webrtc/basics/-toc-disruptive>
- [4] **Real time communication with WebRTC** [online]. <https://codelabs.developers.google.com/codelabs/webrtc-web/-o>
- [5] **Firestore** [online]. <https://firebase.google.com/docs/reference/js/firebase.database.Query?hl=es-419>
- [6] **Ionic** [online]. <https://ionicframework.com/docs/>
- [7] **Angular** [online]. <https://angular.io/guide/quickstart>
- [8] **Crosswalk with Cordova**[online]. <https://crosswalkproject.org/documentation/cordova.html>
- [9] **Angular Google Maps (AGM)** [online]. <https://angular-maps.com/guides/getting-started/>
- [10] **Bootstrap** [online]. <https://v4-alpha.getbootstrap.com/getting-started/introduction/>
- [11] **Tsahi Levent-Levi**. *The Missing chrome://webrtc-internals Documentation* [online]. <https://testrtc.com/webrtc-internals-documentation/>

Anexo A. Firebase

Los datos de un emisor se almacenan en Firebase en formato JSON.

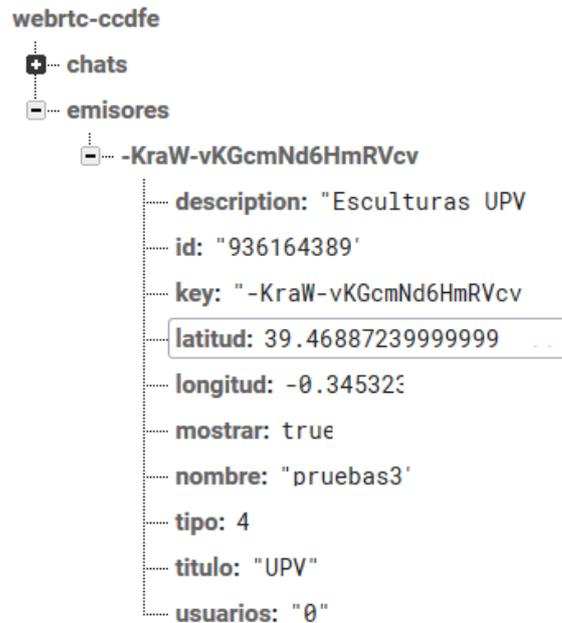


Ilustración 42: Firebase

En la pestaña autenticación de Firebase podemos ver los usuarios que tenemos registrados, desde aquí podemos borrar una cuenta, inhabilitar una cuenta o agregar un nuevo usuario.

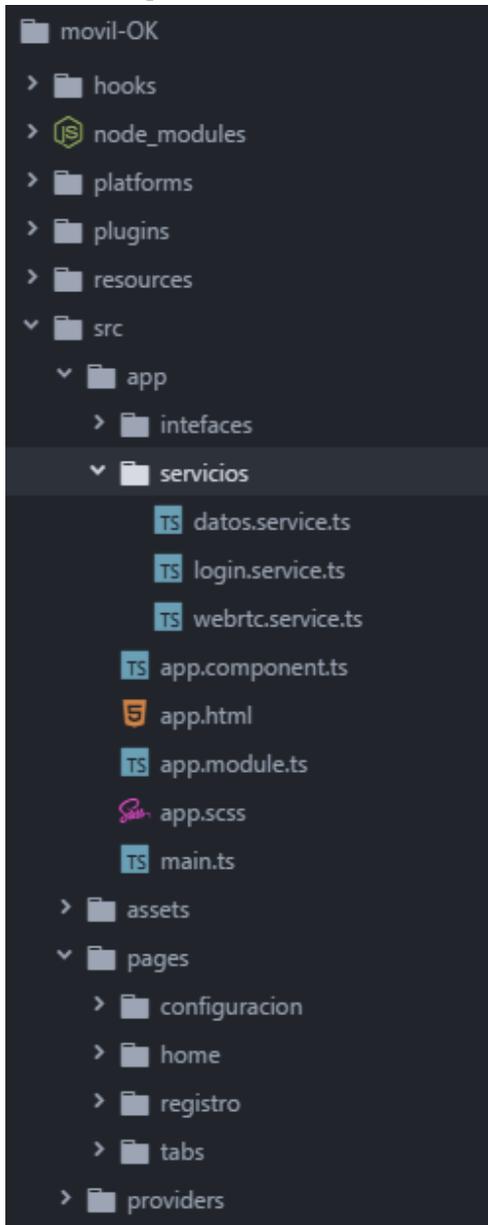
Authentication					CONFIGURACIÓN WEB
USUARIOS	MÉTODO DE ACCESO	PLANTILLAS			
🔍 Buscar por dirección de correo electrónico, número de teléfono o UID de usuario					AGREGAR USUARIO
Identificador	Proveedores	Creado	Accediste a tu cuenta	UID de usuario ↑	
emilijose@mailinator.com	✉️	17 jul. 2017	8 ago. 2017	85YcXgVrSnQEx1mJfv3zBYpVoZH2	
pruebas3@mailinator.com	✉️	1 ago. 2017	12 ago. 2017	DhDQMBcxfhbobsRDHF9LibM2...	
prova@mailinator.com	✉️	17 jul. 2017	17 jul. 2017	TYTgJlfJWjM7LYOueafWcToDxnx1	
pruebas2@mailinator.com	✉️	17 jul. 2017	17 jul. 2017	b72TL605KXcwFqkZ4yEw5dhZicS2	

Filas por página: 50 1 a 4 de 4

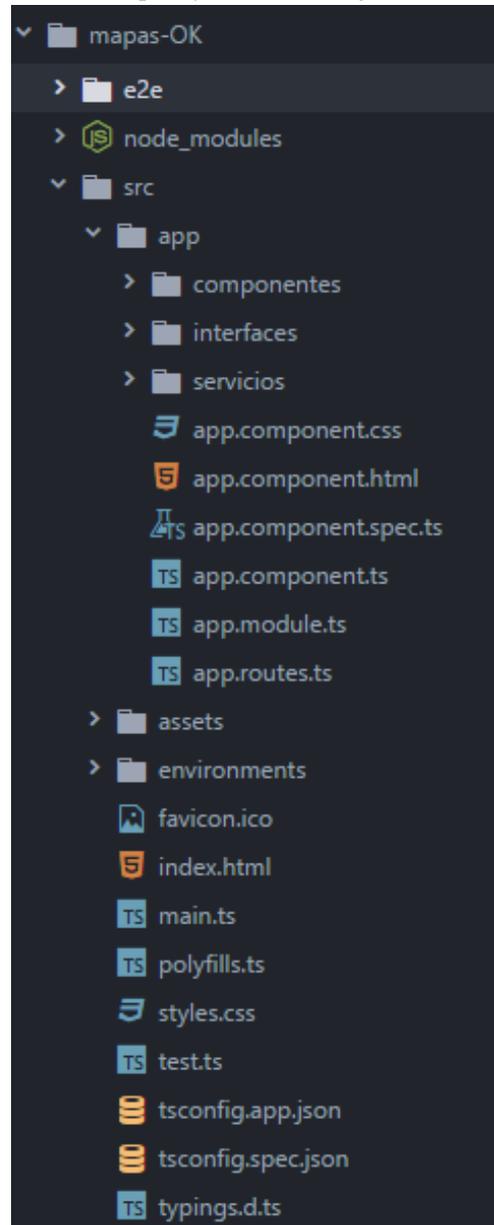
Ilustración 43: Usuarios Firebase

Anexo B. Estructura del proyecto

Estructura aplicación móvil con Ionic 3



Estructura plataforma web, Angular 4



Para tener una mejor organización se ha intentado seguir una misma estructura así pues tenemos en el proyecto las siguientes carpetas:

hooks: esta carpeta contiene scripts que se ejecutan en el proceso de construcción de la aplicación o cuando se produce un determinado evento.

node_modules: están todos los archivos de dependencias descargados por npm, todas las librerías que se declaren como dependencias en el archivo package.json deben estar en esta carpeta.

platforms: contiene el código nativo de la plataforma para la que se va a crear la aplicación y se actualiza al compilar.

plugins: módulos o complementos instalados en la aplicación.

resources: contiene recursos (el icono y las imágenes) que se utilizan en la pantalla de presentación, mientras esperamos que una aplicación arranque.

www: está la versión más reciente del código cuando se efectúa un cambio.

config.xml: este fichero incluye parámetros que se utilizan al construir un proyecto nativo desde proyecto ionic.

package.json: describe las dependencias npm.

tslint.json y tsconfig.json: se utilizan para la compilación mediante el compilador de TypeScript.

src: contiene la implementación del código y es la carpeta donde se realiza el desarrollo.

index.html: es el fichero que se ejecuta al cargar la aplicación y en el que se pueden cargar scripts.

app/interfaces: contiene la interfaz emisor y el fichero de rutas index.ts.

app/servicios: contiene los servicios que se han utilizado datos.service, login.service y webrtc.service.

app/: la carpeta app incluye ficheros como el módulo raíz app.module, y el componente raíz app.component junto con la vista HTML y el CSS.

pages: aquí están todos los componentes utilizados para crear la aplicación móvil, como en el componente raíz cada uno tiene la vista HTML, CSS y la clase controladora.

En la aplicación web realizada con Angular podríamos destacar la carpeta E2E, ya que la estructura es la misma que la aplicación móvil y cambia que los componentes están en la carpeta “componentes” no en la carpeta “pages”.

E2e: esta carpeta contiene los archivos para realizar pruebas unitarias en la aplicación.