



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS  
DE LA COMPUTACIÓN**

**CARRERA DE SISTEMAS E  
INFORMÁTICA**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERO EN SISTEMAS E INFORMÁTICA**

**“MECANISMO DE CONTROL DE DESCONEXIÓN DEL  
SERVICIO DE VIDEOCONFERENCIA MÚLTIPLE EN LA NUBE”**

**AUTORES:**

**Llasag Rosero Raúl Homero  
Luje Pozo Jesús Damián**

**DIRECTOR:**

**Marcillo Parra Diego Miguel**

**Sangolquí**

**19 de enero de 2017**

# CERTIFICADO



DEPARTAMENTO DE CIENCIAS  
DE LA COMPUTACIÓN

CARRERA DE SISTEMAS E INFORMÁTICA

## CERTIFICACIÓN

Certifico que el trabajo de titulación, "MECANISMO DE CONTROL DE DESCONEXIÓN DEL SERVICIO DE VIDEOCONFERENCIA MÚLTIPLE EN LA NUBE" realizado por los señores: RAÚL HOMERO LLASAG ROSERO y JESÚS DAMIÁN LUJE POZO, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar a los señores RAÚL HOMERO LLASAG ROSERO y JESÚS DAMIÁN LUJE POZO para que lo sustenten públicamente.

Sangolquí, 18 de enero de 2017

Ing. Diego Marcillo P. PhD.  
DIRECTOR

## AUTORÍA DE RESPONSABILIDAD



DEPARTAMENTO DE CIENCIAS  
DE LA COMPUTACIÓN

CARRERA DE SISTEMAS E INFORMÁTICA

### AUTORÍA DE RESPONSABILIDAD

Nosotros, RAÚL HOMERO LLASAG ROSERO, con cédula de identidad N° 1724560592 y JESÚS DAMIÁN LUJE POZO, con cédula de identidad N° 1723526016, declaramos que este trabajo de titulación "MECANISMO DE CONTROL DE DESCONEXIÓN DEL SERVICIO DE VIDEOCONFERENCIA MÚLTIPLE EN LA NUBE" ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas. Consecuentemente declaramos que este trabajo es de nuestra autoría, en virtud de ello nos declaramos responsables del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 18 de enero de 2017

Raúl Homero Llasag Rosero  
C.C. 1724560592

Jesús Damián Lujé Pozo  
C.C. 1723526016

## AUTORIZACION



**DEPARTAMENTO DE CIENCIAS  
DE LA COMPUTACIÓN**

**CARRERA DE SISTEMAS E INFORMÁTICA**

### AUTORIZACIÓN

Nosotros, RAÚL HOMERO LLASAG ROSERO y JESÚS DAMIÁN LUJE POZO, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación "MECANISMO DE CONTROL DE DESCONEXIÓN DEL SERVICIO DE VIDEOCONFERENCIA MÚLTIPLE EN LA NUBE" cuyo contenido, ideas y criterios son de nuestra autoría y responsabilidad.

**Sangolquí, 18 de enero de 2017**

**Raúl Homero Llasag Rosero**  
C.C. 1724560592

**Jesús Damián Luje Pozo**  
C.C. 1723526016

## DEDICATORIA

A mi madre Isabel, quien siempre supo cómo alentarme a cumplir cada uno de mis sueños y afrontar mis retos, y cuando no supo, tuvo el tino de encontrar dentro de mí, cada una de mis desalientos. Por mi hermano Diego, pues desde pequeños siempre tuvimos el reto de ser cada día mejores como seres humanos y como emprendedores. Un día con él basta para recargar el espíritu de energía y seguir cumpliendo objetivos.

A mi padre Raúl, le debo mi personalidad intelectual, y lo agradeceré infinitamente. Desde muy pequeño me inculcaba buenas costumbres como hacer ejercicio, leer, escribir, a practicar juegos de concentración como el ajedrez. Junto a mi madre supieron inculcarnos a mí hermano y a mí, que el mejor juguete es el que uno lo crea.

Este trabajo es por ellos y por toda mi familia, pues es una profesión que ha requerido mucho esfuerzo y determinación. Es un pequeño paso para mí, pero lo aprendido hasta este momento se los pongo a su disposición sin ningún egoísmo, pues compartir el conocimiento, solventa las dudas del receptor y engrandece el espíritu del emisor.

Raúl

## **DEDICATORIA**

El presente trabajo le dedico a mis padres y hermano, a mis padres por su total entrega y ser fuente de sabiduría y luz, a mi hermano por ser el compañero de todas mis aventuras.

También les dedico a mis abuelitos y familiares por ser parte de mi formación física, intelectual y espiritual.

Finalmente, lo dedico a mis amigos y compañeros, pues ellos me apoyaron y motivaron en todo momento; unos haciéndome ver el lado hermoso de la vida y, a los otros, por demostrarme el camino que no debía seguir.

Damián

## AGRADECIMIENTO

Estoy infinitamente agradecido con nuestro Director de proyecto: por haberme incluido en el ámbito de la investigación desde mucho tiempo atrás, éste tema investigado tiene gran relación con su tema de tesis doctoral; el presente trabajo se ha logrado gracias a que él, nos ha proveído de una oficina del Departamento de Investigación del DECC y a la Universidad de las Fuerzas Armadas por permitirlo.

A todos mis docentes, amigos y compañeros de Universidad y de aulas, deseo agradecerles por el apoyo brindado en este trabajo. “La conjunción de algunas rocas debidamente ordenadas, hacen posible una edificación”, y es justamente esa conjunción la que quiero agradecer.

Muchas gracias por su aporte.

Raúl

## **AGRADECIMIENTO**

A mis padres, hermano y familiares por ayudarme a potenciar lo mejor de mi ser.

A mis profesores porque me dieron las herramientas necesarias para comprender y valorar lo maravilloso que es la vida.

Al Ing. Diego Marcillo por ser mi mentor en el mundo de la ciencia.

A la Universidad de las Fuerzas Armadas - ESPE.

Damián



## ÍNDICE

<b>CERTIFICADO .....</b>	<b>ii</b>
<b>AUTORÍA DE RESPONSABILIDAD.....</b>	<b>iii</b>
<b>DEDICATORIA .....</b>	<b>v</b>
<b>DEDICATORIA .....</b>	<b>vi</b>
<b>AGRADECIMIENTO .....</b>	<b>vii</b>
<b>AGRADECIMIENTO .....</b>	<b>viii</b>
<b>ÍNDICE.....</b>	<b>ix</b>
<b>ÍNDICE DE TABLAS .....</b>	<b>xi</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>xii</b>
<b>ÍNDICE DE ECUACIONES.....</b>	<b>xiv</b>
<b>GLOSARIO .....</b>	<b>xv</b>
<b>RESUMEN .....</b>	<b>xviii</b>
<b>ABSTRACT .....</b>	<b>xix</b>
<b>DEFINICIÓN DEL PROYECTO.....</b>	<b>1</b>
1.1. <i>Introducción .....</i>	<i>1</i>
1.2. <i>Área de Conocimiento .....</i>	<i>1</i>
1.3. <i>Antecedentes.....</i>	<i>1</i>
1.4. <i>Planteamiento del problema .....</i>	<i>2</i>
1.5. <i>Estado del Arte .....</i>	<i>2</i>
1.6. <i>Objetivos.....</i>	<i>5</i>
1.7. <i>Objetivo General .....</i>	<i>5</i>
1.8. <i>Objetivos Específicos.....</i>	<i>5</i>
1.9. <i>Justificación.....</i>	<i>5</i>
1.10. <i>Alcance .....</i>	<i>6</i>
1.11. <i>Definición de la investigación.....</i>	<i>7</i>
1.12. <i>Hipótesis.....</i>	<i>7</i>
1.13. <i>Herramientas .....</i>	<i>7</i>
1.14. <i>Factibilidad .....</i>	<i>8</i>
<b>ESTADO DEL ARTE Y DE LA CUESTIÓN .....</b>	<b>10</b>
2.1. <i>Introducción .....</i>	<i>10</i>
2.2. <i>Tecnologías de la información sobre la web .....</i>	<i>10</i>
2.3. <i>Servicio de videoconferencia múltiple.....</i>	<i>15</i>
2.4. <i>Arquitectura del Servicio de Videoconferencia.....</i>	<i>19</i>
2.5. <i>Estudios de Calidad .....</i>	<i>27</i>
2.6. <i>TRABAJOS RELACIONADOS.....</i>	<i>32</i>
<b>DISEÑO, DESARROLLO, IMPLEMENTACIÓN, IMPLANTACIÓN.....</b>	<b>36</b>
3.1. <i>Introducción.....</i>	<i>36</i>
3.2. <i>Componentes elementales de una videoconferencia .....</i>	<i>36</i>
3.3. <i>Plataforma base del servicio de videoconferencia en la nube.....</i>	<i>37</i>
3.4. <i>Análisis y diseño de la arquitectura basado en patrones .....</i>	<i>38</i>
3.5. <i>Metodología XP en la plataforma base de videoconferencia.....</i>	<i>40</i>
3.6. <i>Problemas a causa de interrupciones del canal de comunicación .....</i>	<i>53</i>
3.7. <i>Mecanismo de control de desconexión .....</i>	<i>54</i>

	x
3.8. <i>Análisis y Diseño del mecanismo basado en patrones</i> .....	54
3.9. <i>Metodología XP en el mecanismo de control de desconexión</i> .....	56
3.10. <i>Implantación y herramientas base</i> .....	64
<b>PRUEBAS DE CALIDAD</b> .....	<b>66</b>
4.1. <i>Introducción</i> .....	66
4.2. <i>Evaluación de calidad en los servicios de streaming</i> .....	66
4.3. <i>Definición de casos de prueba de calidad de servicio</i> .....	67
4.4. <i>Modelo Matemático con variables de tiempo</i> .....	68
4.5. <i>Estándares de Evaluación y valores aceptables</i> .....	72
4.6. <i>Entorno</i> .....	73
4.7. <i>Casos de Evaluación de videoconferencia y análisis de resultados de pruebas de calidad de servicio</i> .....	73
4.8. <i>Casos de Evaluación de videoconferencia múltiple y análisis de resultados de calidad de servicio</i> .....	79
4.9. <i>Análisis de variación de parámetros QoS</i> .....	83
4.10. <i>Experiencia de usuarios en la plataforma</i> .....	87
4.11. <i>Resultados de las pruebas de calidad de experiencia</i> .....	89
<b>CAPÍTULO V</b> .....	<b>96</b>
<b>CONCLUSIONES Y TRABAJOS FUTUROS</b> .....	<b>96</b>
5.1. <i>Conclusiones</i> .....	96
5.2. <i>Trabajos futuros</i> .....	97
<b>BIBLIOGRAFÍA</b> .....	<b>98</b>

## ÍNDICE DE TABLAS

Tabla 1 Hardware a usarse en el desarrollo del proyecto .....	8
Tabla 2 Software a usarse en el desarrollo del proyecto .....	8
Tabla 3 Comparación de TCP, UDP y SCTP .....	26
Tabla 4 Mean Opinion Score MOS .....	32
Tabla 5 Historia de usuario - Ingreso al sistema .....	41
Tabla 6 Historia de usuario - Contactar usuario .....	41
Tabla 7 Historia de usuario - Sala de videoconferencia común .....	41
Tabla 8 Descripción del caso de uso - Ingreso .....	42
Tabla 9 Descripción del caso de uso - Contacto.....	43
Tabla 10 Descripción del caso de uso - Videoconferencia múltiple .....	44
Tabla 11 Control de iteraciones del desarrollo de videoconferencia múltiple .....	49
Tabla 12 Caso de prueba - Registro de usuario .....	50
Tabla 13 Caso de prueba - Ingreso al sistema .....	50
Tabla 14 Caso de prueba - Contactar usuario.....	51
Tabla 15 Caso de prueba - Sala de videoconferencia común .....	52
Tabla 16 Historia de usuario - Reconectar llamada.....	56
Tabla 17 Historia de usuario - Grabar audio y video .....	56
Tabla 18 Historia de usuario - Recuperar información.....	57
Tabla 19 Descripción de caso de uso - Restablecer llamada.....	57
Tabla 20 Control de iteraciones del mecanismo de control de desconexión.....	61
Tabla 21 Caso de prueba - Restablecer llamada.....	62
Tabla 22 Caso de prueba - Grabación de audio y video .....	63
Tabla 23 Caso de prueba - Visualizar audio y video .....	64
Tabla 24 Casos de evaluación .....	67
Tabla 25 Definición de variables de tiempo .....	70
Tabla 26 Parámetros de calidad de audio .....	72
Tabla 27 Parámetros de calidad de video .....	72
Tabla 28 Equipos usados en las pruebas de calidad .....	73
Tabla 29 Audio y Video de iMac en el caso CE001 .....	76
Tabla 30 Audio y Video en NUC en el caso CE001 .....	76
Tabla 31 Parámetros en tiempo real del módulo de estadísticas .....	77
Tabla 32 Audio y video de iMac en el caso CE002 .....	79
Tabla 33 Audio y Video de NUC en el caso CE0002 .....	79
Tabla 34 Audio y Video de iMac que desconecta de NUC en el caso CE003 .....	82
Tabla 35 Audio y Video de NUC que detecta a iMac en el caso CE003 .....	82
Tabla 36 Audio y Video de iMac que desconecta de iMac 2 en el caso CE003 .....	82
Tabla 37 Audio y Video de iMac 2 que detecta a iMac en el caso CE003 .....	83
Tabla 38 Estadísticas de Audio y Video en videoconferencia con 6 usuarios .....	94

## ÍNDICE DE FIGURAS

Figura 1 El mundo real .....	13
Figura 2 Uso de STUN para obtener IPs públicas .....	14
Figura 3 Uso de Servidores STUN y TURN .....	15
Figura 4 Clasificación de los sistemas de computación .....	20
Figura 5 Conjunto de protocolos en la Arquitectura TCP/IP .....	24
Figura 6 Manejo de un flujo de paquetes .....	28
Figura 7 Manejo de un excesivo jitter .....	30
Figura 8 Arquitectura con Realidad Aumentada .....	33
Figura 9 Arquitectura Automatic Reconnection WebRTC (ARW) .....	34
Figura 10 Sistema Distribuido P2P híbrido .....	37
Figura 11 Arquitectura MVC .....	39
Figura 12 Arquitectura MVC de Videoconferencia .....	40
Figura 13 Caso de uso - Ingreso a la plataforma .....	42
Figura 14 Caso de uso - Contacto.....	43
Figura 15 Caso de uso - Videoconferencia múltiple .....	43
Figura 16 Modelo de BD .....	44
Figura 17 Diagrama de Clases de videoconferencia .....	45
Figura 18 Alice llama a Bob.....	46
Figura 19 Ciclo de vida de una videoconferencia con tres usuarios .....	47
Figura 20 Diagrama de secuencia de videoconferencia múltiple .....	48
Figura 22 Caso de prueba – Contactar usuario 2.....	51
Figura 23 Caso de prueba – Sala de videoconferencia común .....	52
Figura 24 Arquitectura del mecanismo de control de desconexión.....	55
Figura 25 Caso de uso - Reestablecer llamada .....	57
Figura 26 Diagrama de clases de mecanismo de control.....	58
Figura 27 Reconexión de una llamada de dos usuarios.....	59
Figura 28 Reconexión de una videoconferencia múltiple .....	60
Figura 29 Caso de prueba - Reconectar llamada - Usuario desconectado.....	62
Figura 30: Caso de prueba - Visualizar audio y video .....	63
Figura 31 Diagrama de implantación .....	65
Figura 32 Ciclo de vida sin desconexiones .....	68
Figura 33 Ciclo de vida con dos desconexiones.....	69
Figura 34 RTT en CE001 .....	74
Figura 35 iMac desconectado en CE001 .....	75
Figura 36 NUC que detecta desconexión en CE001 .....	75
Figura 37 iMac desconectado en CE002 .....	78
Figura 38 NUC que detecta desconexión en CE002 .....	78
Figura 39 iMac desconectado en CE003 con NUC.....	80
Figura 40 NUC que detecta desconexión en CE003 de iMac .....	81
Figura 41 iMac desconectado en CE003 con iMac 2 .....	81
Figura 42 iMac 2 que detecta desconexión en CE003 de iMac.....	81
Figura 43 Variación del Retraso de audio en CE003 .....	84
Figura 44 Variación del Jitter Recibido de audio en CE003 .....	84
Figura 45 Variación del almacenamiento JitterBuffer de audio en CE003 .....	85
Figura 46 Variación de la pérdida de paquetes de audio en CE003 .....	85
Figura 47 Variación de la Retraso de video en CE003.....	86
Figura 48 Variación del almacenamiento de jitterBuffer de video en CE003.....	86

	xiii
Figura 50 Recuento de calidad de escucha en videoconferencia.....	89
Figura 51 Recuento de calidad de escucha en recuperación .....	89
Figura 52 MOS de la calidad de escucha .....	90
Figura 53 Recuento de esfuerzo de escucha en videoconferencia.....	91
Figura 54 Recuento de esfuerzo de escucha en recuperación .....	91
Figura 55 MOS de la calidad de escucha .....	92
Figura 56 Recuento de calidad de video en videoconferencia .....	92
Figura 57 Recuento de calidad de video en recuperación .....	93
Figura 58 MOS de la calidad de video .....	93

## ÍNDICE DE ECUACIONES

Ecuación 1 Retraso .....	29
Ecuación 2 Conexión.....	70
Ecuación 3 Desconexión .....	70
Ecuación 4 Tiempo perdido por hardware .....	70
Ecuación 5 Tiempo fuera de websocket .....	70
Ecuación 6 Reconexión .....	70
Ecuación 7 Grabación.....	70
Ecuación 8 Total de segundos perdidos .....	70
Ecuación 9 Primera fórmula generalizada del tiempo total.....	70
Ecuación 10 Segunda fórmula generalizada del tiempo total.....	71
Ecuación 11 Tercera fórmula generalizada del tiempo total .....	71
Ecuación 12 bytes a Megabits .....	94

## GLOSARIO

### A

**API:** conjunto de métodos, protocolos y herramientas para construir software. (Orenstein, 2010)

### B

**Bootstrap:** un framework para el desarrollo rápido y fácil de web en móviles. (Bootstrap, 2011)

### C

**CSS:** es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla (W3C, 2016)

### D

**Disrupción:** Disrupción: rotura o interrupción brusca (RAE, 2016)

### F

**Framework:** Desde el punto de vista de desarrollo web, es algo que sirve para su desarrollo como: aplicaciones, sitios dinámicos y servicios. Proporcionan herramientas, bibliotecas, plantillas, códigos y ejemplos de aplicaciones. (ALEGSA, 2016)

### H

**HTML5:** la última versión de HTML que incorpora soporte para conectividad y multimedia. (Mozilla Foundation, 2016)

**HTTP:** protocolo utilizado en las comunicaciones en Internet (Gourley, 2002)

**HTTPS:** versión segura de HTTP, que utiliza criptografía y certificados para su funcionamiento. (Gourley, 2002)

### I

**IANA:** entidad que supervisa y controla la asignación de direcciones IP, servidores raíz de DNS y otros recursos de Internet. (IANA, 2016)

**IETF:** organización encargada de producir documentos de alta calidad y relevancia técnica relacionados al diseño, uso y manejo de Internet. (IETF, 1992)

**ITU/UIT:** un organismo especializado de las Naciones Unidas para las Tecnologías de la Información y Comunicación que elabora normas técnicas que garantizan la interconexión de redes y tecnologías. (ITU, 2016)

## J

**Javascript:** lenguaje de desarrollo de aplicaciones cliente/servidor a través de Internet. (Sánchez, 2012)

## P

**Peer to Peer:** es un sistema auto-organizado de entidades iguales y autónomas que comparten el uso de recursos distribuidos en un ambiente de red. (Steinmetz & Wehrle, 2005)

**Python:** es un lenguaje de programación interpretado, interactivo y orientado a objetos. (Python Org., 2001)

## Q

**QoE:** Calidad de experiencia, calidad del proceso de comunicación según es percibido por el usuario. (Bonini, 2012)

**QoS:** un conjunto de tecnologías que le proporcionan la capacidad para administrar el tráfico de red de manera rentable y mejorar las experiencias de usuario (Microsoft, 2012)

## R

**RTC:** cualquier comunicación en vivo que ocurre con una latencia mínima. (Real Time Communications, 2016)



**S**

**Streaming:** Técnica de transferencia de datos, de manera tal, que puedan ser procesados como una corriente continua y uniforme. (Clark & Clark, 2004)

**T**

**TCP:** protocolo de enrutamiento que asegura un servicio fiable, orientado a la conexión. (Atelin, 2007)

**U**

**Umbral:** Es el valor más bajo de una magnitud a partir del cual un fenómeno se vuelve perceptible (K Dictionaries Ltd., 2013).

**URL:** localizador uniforme de recursos que permite localizar de forma sencilla cualquier recurso de red. (Uca, 2016)

**W**

**WebM:** formato abierto de video diseñado para la Web, define la estructura del contenedor conteniendo streams comprimidos de VP8 y VP9. (WebM Project, 2010)

**WebRTC:** es un proyecto libre que proporciona a los navegadores y aplicaciones móviles con comunicaciones en tiempo real (RTC) a través de las API simples. (WebRTC, 2011)

**W3C:** comunidad internacional que desarrolla protocolos y directrices que aseguren el crecimiento de la Web a largo plazo. (W3C, 2016)

## RESUMEN

El creciente avance de la tecnología y la necesidad del ser humano de comunicarse continuamente con sus semejantes, da origen al desarrollo de nuevas tecnologías que permitan conectar a otras personas independientemente de su ubicación y con tiempos de latencia bajos que permitan una comunicación fluida. La calidad de ésta interacción puede verse afectada por múltiples factores que atenten contra la continuidad de la transmisión de la información, provocando interrupciones que generan malestar en los interlocutores y pérdida de la información. WebRTC es un framework que permite la comunicación entre los usuarios, tanto para video, audio y datos, utilizando únicamente un navegador web y la conexión a Internet. Sus usuarios se conectan directamente entre sí utilizando un servidor únicamente para el establecimiento inicial y la negociación de la comunicación. Sin embargo WebRTC no es capaz de restablecer la comunicación en caso de una desconexión, tal como otros sistemas de videoconferencia que carecen de un control de desconexiones en las comunicaciones en tiempo real, y de una forma de recuperar la información que debió ser transmitida entre los usuarios afectados, especialmente cuando se presenta el complejo escenario en el que existen múltiples interlocutores afectados por la misma. El presente proyecto utiliza el framework WebRTC para el establecimiento de la comunicación entre varias conexiones P2P y propone un mecanismo para el control de las desconexiones, con el fin de reducir la información perdida durante la desconexión, enfocándose en el escenario en el que varios usuarios interactúen en una sala de videoconferencia común.

### **Palabras clave:**

- **STREAMING**
- **WEBRTC**
- **DISRUPCIÓN**
- **P2P**

## **ABSTRACT**

The increasing advance of the technology and the need of the human to communicate continuously with its like gives origin to the development of new technologies that allow to connect to other people irrespective of the place in which they are and with low times of latency that allow a fluid communication. The quality of this interaction can be affected by multiple factors that influence to the continuity of the transmission of information between users, causing disruptions that generate discomfort of consumers and loss of information. WebRTC is an open framework that allows communication between users, both for video, audio and data, using only a web browser and Internet connection. On this, your users can connect directly to each other using a server only for initial setup and negotiation of communication. But, WebRTC can't be reestablish communication, when a disconnection to the network appears. Other videoconferencing systems too lack a control of disconnections in real-time communications and a way to recover the information that should have been transmitted between the users affected by such disruption, especially when presenting the complex scenario with multiple partners affected by the same disruption. The present project uses the WebRTC framework to establish the communication between P2P connections and proposes a mechanism for the control of the disconnections, in order to reduce the information lost during the disconnection, focusing on the scenario in which several users interact in a common videoconference room.

### **Keywords:**

- **STREAMING**
- **WEBRTC**
- **DISRUPCIÓN**
- **P2P**

# CAPÍTULO I

## DEFINICIÓN DEL PROYECTO

### 1.1. Introducción

El presente capítulo describe el problema encontrado en los servicios de videoconferencia al presenciar desconexiones de red, los antecedentes que llevaron a descubrirlo y el alcance de los objetivos a cumplir en el presente proyecto, con los que gracias al uso de las herramientas adecuadas se pretende cumplir con ésta investigación, para finalmente aceptar o rechazar la hipótesis planteada.

### 1.2. Área de Conocimiento

El presente proyecto está orientado a la línea de investigación de Tecnologías de la Información y Comunicación, de acuerdo a las actividades propuestas por la Universidad de las Fuerzas Armadas - ESPE para contribuir con el desarrollo sustentable del país; enfocándose en las áreas de: Protocolos de Comunicación y Sistemas Distribuidos.

### 1.3. Antecedentes

La necesidad de aplicaciones que permitan la transmisión multimedia en tiempo real nace para acortar tiempos de respuesta en negocios, comercio electrónico, gobierno electrónico y en la educación que dependía de servicios de comunicación bajo licencia (Granda, 2015). La solución a esta problemática ha sido posible gracias al framework libre Web Real-Time Communication (WebRTC) (WebRTC, 2011) y al conjunto de componentes de alta calidad de red, audio y video que se implementan en un navegador sin la necesidad de instalar complementos, mediante el uso de las Application Programming Interface (APIs) del lenguaje de programación de JavaScript, como también a las tecnologías libres para la implementación de aplicaciones web como son HyperText Markup Language versión 5 (HTML5), HyperText Transfer Protocol (HTTPS) y Transmission Control Protocol (TCP/IP).

WebRTC trabaja con el protocolo de comunicación punto a punto p Peer to Peer (P2P), en donde los puntos se comportan como iguales, es decir pueden ser clientes o servidores a la vez. Los equipos involucrados en la transferencia de información

pueden diferir en su configuración, como son: la velocidad de proceso, ancho de banda, incluso el estado de la conexión a la Red (Gatica, 2005).

Dado que el estado de conexión puede variar, surgen, como la pérdida de información en este tipo de comunicaciones. En la Universidad de las Fuerzas Armadas ESPE se desarrolló un proyecto (Yacchirema & Ortega, 2015) que pretendía mitigar la pérdida de datos en una video llamada P2P mediante la implementación de un algoritmo de reconexiones de sesión que le permitiera mantener su estado.

#### **1.4. Planteamiento del problema**

En los canales de comunicación de una videoconferencia múltiple desarrollada sobre el Proyecto de marco abierto WebRTC, se pueden producir interrupciones a lo largo del tiempo, debido a las desconexiones de red. Hasta el momento no se ha logrado implementar un mecanismo que analice el estado actual de cada punto para controlar la reconexión de la sesión.

En el instante en que uno de los usuarios partícipes de una videoconferencia múltiple entra en el proceso de desconexión, a éste automáticamente se le elimina la suscripción del Streaming en cuestión, perdiendo información de la llamada. La comunicación no se restablece, aún si el estado de conexión con la Red se haya recuperado.

La presencia continua de interrupciones incrementa la cantidad de información no transmitida, generando descontento al usuario, puesto que la calidad de experiencia (QoE) con una plataforma de videoconferencia múltiple desarrollada sobre WebRTC, disminuye.

#### **1.5. Estado del Arte**

El Internet fue diseñado originalmente para soportar comunicación de datos, siendo en su mayoría emails y archivos. El tráfico de voz era limitado exclusivamente a redes telefónicas. Con el enorme crecimiento de usuarios y aplicaciones en Internet, nació la necesidad de comunicación multimedia que dio origen a nuevas arquitecturas y protocolos que permitan satisfacer nuevas demandas. (Durresti, 2005)

Las demandas identificadas se clasifican en dos tipos: Aplicaciones de streaming no interactivas y Aplicaciones interactivas en tiempo real. En el primer caso se cuenta con un archivo almacenado que será transmitido bajo demanda. En segundo caso los datos son generados y transmitidos en tiempo real (Durresi, 2005).

Anteriormente para visualizar el contenido bajo demanda y en tiempo real se requería del uso de extensiones de terceros para el navegador, como Adobe Flash o Microsoft Silverlight (Adobe Systems Incorporated, 2016). Hoy en día los navegadores web soportan nativamente ciertas funcionalidades como HTML5 y Cascading Style Sheets (CSS3), que posibilitan la reproducción de audio y video. (Burkhard Stiller, 2014).

WebRTC es un proyecto libre que permite a los navegadores realizar comunicación en directo, sin el uso de extensiones adicionales en el navegador, además de la transmisión de audio, video y datos con baja latencia para servicios de comunicación en tiempo real y bajo demanda. Para ello WebRTC provee de APIs que facilitan el manejo de sus diferentes características (Burkhard Stiller, 2014).

WebRTC es auspiciado por Google, Mozilla y Opera, y en su especificación maneja los siguientes conceptos:

- **MediaStream:** Permite el manejo de los flujos de audio y vídeo desde fuentes como el micrófono y la cámara del usuario (W3C, 2016).
- **RTCPeerConnection:** Responsable del establecimiento y manejo apropiado de las conexiones punto a punto (W3C, 2016).
- **RTCDataChannel:** Posibilita la transmisión de cualquier tipo de dato de forma bidireccional entre los puntos (W3C, 2016).

La fundamentación de este proyecto se basa principalmente en recomendaciones de estudios previos de (Yacchirema & Ortega, 2015), que han realizado un estudio comparativo de las principales APIs y Codificadores / Decodificadores (CODECs) de Video en la tecnología WebRTC, a continuación se desarrolló e implantó un mecanismo de control de desconexiones en comunicaciones punto a punto, el cual se limita

únicamente a dos usuarios concurrentes, enfocándose en su uso sobre dispositivos móviles con Sistema Operativo Android.

- **Metodología de Desarrollo Ágil Extreme Programming (XP):** Es una metodología propuesta por Kent Beck, utilizada para desarrollar software con alta calidad de la manera más rápida posible y con el mayor beneficio para el cliente. Se caracteriza por tener ciclos de desarrollo breves, integración constante, retroalimentación continua por parte del cliente y enfoque de equipo (Wells, 2013).

La metodología XP recomienda la programación en parejas que a pesar de que puede retrasar el proceso, asegura que muchos de los errores sean detectados conforme son introducidos en el código (Wells, 2013). El ciclo de vida de la metodología consiste en seis fases:

- Exploración: Se plantean de forma general las historias de usuario, familiarización de las herramientas y arquitectura del sistema.
- Planificación de Entrega: Se estima el esfuerzo necesario para la entrega puntual.
- Interacciones: Se realizan iteraciones sobre el sistema antes de ser entregado, trabajo realizado sobre las historias de usuario.
- Producción: Pruebas adicionales y revisiones de rendimiento antes de ser trasladado al entorno operativo.
- Mantenimiento: Sistema en producción y se realizan iteraciones de cambio sobre el sistema.
- Muerte del proyecto: Culmina el proyecto si no existen mas historias de usuario por cumplir.

## **1.6. Objetivos**

### **1.7. Objetivo General**

Desarrollar e implantar un mecanismo de control de desconexión sobre una plataforma del servicio videoconferencia múltiple en la nube, utilizando APIs de WebRTC que permitan mitigar las interrupciones del canal de comunicación.

### **1.8. Objetivos Específicos**

- Analizar tecnologías y protocolos para el servicio de videoconferencia en la nube.
- Diseñar e Implementar una plataforma de videoconferencia múltiple en la nube, utilizando la metodología de desarrollo Ágil XP
- Evaluar los problemas que genera la interrupción del canal de comunicación en el servicio de videoconferencia múltiple.
- Diseñar y desarrollar el mecanismo de control de desconexión basado en patrones de diseño.
- Implantar el mecanismo en el servicio de videoconferencia en la nube, utilizando la metodología de desarrollo Ágil XP
- Realizar pruebas funcionales e interpretar los resultados de las pruebas sobre el aplicativo valorando la calidad de servicio (QoS) y experiencia (QoE).

### **1.9. Justificación**

Hoy en día el 35% de los 7 mil millones de personas del mundo están constantemente conectadas a Internet (3CX, 2016) mediante tabletas, teléfonos inteligentes y computadores. Su uso está creciendo drásticamente, al igual que la demanda de servicios de comunicación de mayor calidad, es ahí donde se nace WebRTC.

WebRTC es una tecnología que desde su lanzamiento por Google en 2011 (Burkhard Stiller, 2014) ha venido ganando gran atención dentro de las telecomunicaciones (Emmanuel Bertin, 2015). Según (3CX, 2016), el servicio de videoconferencia que se puede brindar mediante ésta, ayuda a cerrar más ventas, gracias al beneficio de comunicación cara a cara.



Por otra parte (Transparency Market Research, 2016) prevé un crecimiento de 8,5% durante el período 2015-2023 en equipos y servicios de videoconferencia. Sin embargo, las empresas que ofertan el servicio de videoconferencia no se han preocupado de solucionar los problemas causados debido a las interrupciones a las que está expuesta su arquitectura de comunicación punto a punto.

Como solución al problema identificado, la presente investigación mantuvo la idea de desarrollar una plataforma de videoconferencia múltiple sobre el framework libre WebRTC, que controle la reconexión automática de una videoconferencia y mitigue la pérdida de información a causa de las interrupciones que afectan al canal de comunicación.

#### **1.10. Alcance**

En presente proyecto ha incluido un análisis de tecnologías y protocolos para el servicio de videoconferencia. Con el uso de la metodología Ágil XP y patrones de arquitectura, se ha procedido con el diseño e implementación de una plataforma de videoconferencia múltiple en la nube, con el objetivo de identificar las diferentes problemáticas, causadas por interrupciones en los canales de comunicación sobre la tecnología usada.

Posteriormente se ha diseñado, desarrollado e implantado el mecanismo de control de desconexión sobre el servicio de videoconferencia múltiple construido anteriormente. Finalmente se ha realizado un análisis de las pruebas funcionales del aplicativo valorando QoS y QoE, para concluir con un escenario ideal, en el cual la comunicación es óptima.

El presente proyecto se ha desarrollado sobre la tecnología WebRTC, enfocada en la compatibilidad con el Navegador Google Chrome, el cual es multiplataforma y está en continuos cambios por sus autores (Google), pudiendo presentar actualizaciones durante el transcurso del proyecto. Sin embargo estas mejoras no han afectado al desarrollo de la investigación.

### **1.11. Definición de la investigación**

Según (Sabino, 2008) llamamos investigación científica, de un modo general, a la actividad que nos permite obtener conocimientos científicos, es decir, conocimientos que se procura sean objetivos, sistemáticos, claros, organizados y verificables.

Según (Dávila Neuman, 2006) El método inductivo es de datos particulares llegar a una conclusión general, basándose en la utilización de herramientas de inferencia para lograrlo. (Sabino, 2008) menciona que la investigación aplicada persigue fines más directos e inmediatos; ésta recibe el nombre de investigación práctica o empírica que se caracteriza porque busca la aplicación o utilización de los conocimientos adquiridos.

El propósito de la investigación en este proyecto es una investigación aplicada. Según (Murillo, 2008) en la investigación experimental se manipula una o más variables de estudio, de esta manera se pueden controlar el aumento o disminución de esas variables y su efecto en las conductas observadas. Para efecto de la presente tesis, la metodología es experimental, dado el hecho que se requiere realizar pruebas de calidad para dar una valor al servicio.

### **1.12. Hipótesis**

H0: Un mecanismo de control de desconexión en el servicio de videoconferencia reduce la pérdida de información de un punto de conexión, cuando se presentan disrupciones en el canal de comunicaciones a causa de las desconexiones en la Red.

H1: Un mecanismo de control de desconexión en el servicio de videoconferencia múltiple no reduce la pérdida de información de cualquiera de los puntos de conexión, cuando alguno presenta disrupciones en el canal de comunicaciones a causa de las desconexiones en la Red.

### **1.13. Herramientas**

Las herramientas que fueron utilizadas en el presente proyecto se detallan a continuación:

**Tabla 1**  
**Hardware a usarse en el desarrollo del proyecto**

Equipo	Características
Intel Nuc1	Intel core i5 Memoria Ram 4GB Almacenamiento 500GB
Intel Nuc2	Intel core i5 Memoria Ram 4GB Almacenamiento 500GB
iMac	Intel core i5 Memoria Ram 4GB Almacenamiento 1000GB
iMac 2	Intel core i5 Memoria Ram 4GB Almacenamiento 1000GB

**Tabla 2**  
**Software a usarse en el desarrollo del proyecto**

Componente	Aplicaciones a Usar
Navegador	Google Chrome: versión mínima 48
Servidor Web	Apache 2
Base de Datos	MySQL 5.7
Edito de texto	Sublime Text 3
Servidor WebSockets	Interprete Python 3.5.1

#### 1.14. Factibilidad

##### Técnica

Para el desarrollo del proyecto se ha contado con los suficientes recursos de hardware, pertenecientes a la Universidad de las Fuerzas Armadas ESPE.

##### Económica

El proyecto ha sido viable gracias al aporte económico de los titulares de la presente investigación, cubriendo gastos en software y recursos humanos necesarios para el desarrollo e implantación del mismo, los Recursos de Hardware, que son propiedad de la Universidad de las Fuerzas Armadas ESPE, han sido puestos a disposición a los investigadores sin ninguna restricción de horario.

##### Operativa

El proyecto es viable gracias al apoyo del Área de Investigación del Departamento de Ciencias de la Computación, quienes brindaron de un espacio físico, recursos tecnológicos y asesoría técnica para el desarrollo e implantación del caso de estudio. La

Asesoría Técnica ha sido brindada por parte del PhD Diego Marcillo y la PhD Tatiana Gualotuña, docentes tiempo completo de la Universidad de las Fuerzas Armadas ESPE, quienes, con su experiencia en sus tesis doctorales de la Universidad de las Palmas de Gran Canaria, motivaron el trabajo de investigación del presente tema.

## **CAPÍTULO II**

### **ESTADO DEL ARTE Y DE LA CUESTIÓN**

#### **2.1. Introducción**

El presente capítulo contiene fundamentos histórico-teóricos de conceptos, tecnologías y avances en el desarrollo de la API de WebRTC en cuanto a la mejora de la calidad de sus componentes y su aporte al servicio de videoconferencia que ofrece a la comunidad de Internet. También se presentan trabajos de investigación que pretendan mitigar pérdidas de información a causa de las desconexiones de red inalámbrica en los servicios de comunicaciones en tiempo real.

#### **2.2. Tecnologías de la información sobre la web**

##### **2.2.1. HTML**

HTML es un Lenguaje de marcas que representa textos con estructura y formato sobre un navegador. Éste es usado para construir la capa de presentación de una aplicación Web, por lo tanto sus etiquetas representan una estructura como: títulos, subtítulos, tablas, etc. O de formato como: negrilla, subrayado, centrado, justificado, etc.

##### **2.2.2. HTML5**

HTML5 es la última versión del Lenguaje HTML, diseñada para que todos los desarrolladores de la web abierta puedan utilizar. Trae consigo nuevos elementos, atributos y comportamientos. Además provee de un más amplio conjunto de tecnologías que permiten a los sitios web y a las aplicaciones ser de mayor alcance.

##### **2.2.3. Navegador Web**

Un navegador web es una aplicación de software que se ejecuta en la máquina cliente y realiza peticiones HTTP hacia un servidor web, recibiendo respuestas generalmente en lenguaje HTML y presentando al usuario el contenido de acuerdo a las marcas estandarizadas.

#### **2.2.4. Servidor Web**

Un servidor web, es una aplicación de software que entiende y trabaja con el protocolo HTTP. En base al protocolo TCP, un servidor Web abre un puerto de comunicaciones por el cual recibe conexiones de sus clientes. Éste servidor incluye la lógica de programación capaz de gestionar las peticiones específicas del protocolo HTTP.

#### **2.2.5. WebRTC**

WebRTC como proyecto libre, proporciona a los navegadores y aplicaciones móviles de comunicación en tiempo real a través de sus APIs y su conjunto de normas, permitiendo una comunicación a través de un conjunto común de protocolos.

No depende de complementos o software propietario, convirtiendo la comunicación en tiempo real en bondad para cualquier aplicación web a través de una sencilla API de Javascript. Ofrece aplicaciones RTC como la videoconferencia y el intercambio de datos entre navegadores, sobre el modelo punto a punto.

El proyecto WebRTC está en construcción activa tanto a nivel de API de navegador por el World Wide Web Consortium (W3C), como en los niveles de transporte y de protocolos por la Internet Engineering Task Force (IETF). Abstrae su complejidad en tres APIs que se describen a continuación y ofrece apoyo a todos sus usuarios sobre las mismas. (NTT Communications, 2015)

##### **2.2.5.1. APIs**

#### **MediaStream**

Esta interfaz representa un flujo de contenido de medios. Un objeto MediaStream puede recoger cero o más interfaces MediaStreamTrack (pista individual), que son una pista de audio o de video (Mozilla, 2016). Si se requiere adquirir flujos de audio y video desde una plataforma subyacente se utiliza la API getUserMedia(), la cual optimiza, codifica y decodifica, gracias a los motores de audio y video de WebRTC (Grigorik, 2013). Finalmente para una salida MediaStream, se toma en cuenta que esta se puede enviar a uno o más destinos mediante código JavaScript para el post-procesamiento.

## **RTCPeerConnection**

Esta interfaz representa la conexión WebRTC real, permite establecer comunicaciones entre el equipo local y un par remoto (Mozilla, 2016). RTCPeerConnection es responsable de la gestión del ciclo de vida de cada conexión punto a punto y encapsula la configuración de la conexión, la gestión y su estado dentro de una única interfaz (NTT Communications, 2015).

En el momento en que una persona llama deseando iniciar una conexión con una ubicación remota, un objeto RTCPeerConnection se instancia y tiene un estado de señalización, un estado de recolección ICE y un estado de conexión ICE. Incluye una descripción autogenerada SDP para intercambiar con el destinatario, el cual responde con su propia descripción SDP. Cuando la conexión se establece, “RTCPeerConnection permite él envió de datos de audio y video en tiempo real como un flujo de bits entre los navegadores” (NTT Communications, 2015).

## **RTCDataChannel**

Esta interfaz representa un canal de red que puede ser utilizado para las transferencias bidireccionales punto a punto de datos arbitrarios. Cada canal está relacionado con un RTCPeerConnection, el cual puede poseer varios canales de datos, en teoría puede llegar a tener 65.534, variando de un navegador a otro (Mozilla, 2016).

RTCDatachannel API se asemeja a WebSocket porque es un superconjunto de su API, con la diferencia en que ésta interfaz ofrece transporte subyacente y fue creada para transmitir datos sobre el modelo punto a punto, más no orientada a la conexión con un servidor. DataChannel se puede configurar para entregar la misma fiabilidad y garantías en orden de aviso como WebSocket. Aunque el poder real de DataChannel se debe precisamente al hecho de que no tiene que seguir el orden de entrega fiables y la semántica (Grigorik, 2013).

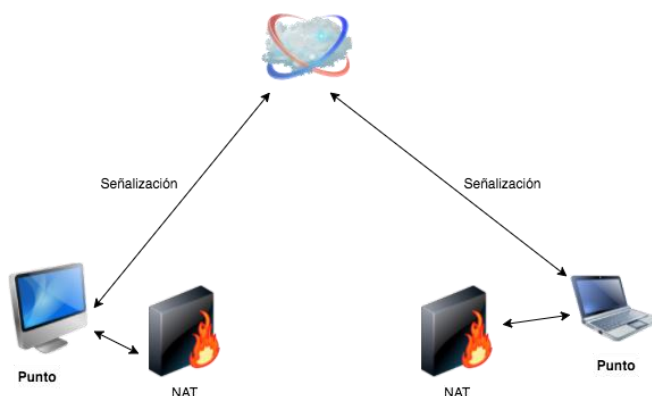
Un Datachannel puede ser configurado para entregar confiable o parcialmente confiable los mensajes, o también dentro o fuera de orden de la secuencia de mensajes enviados. Cada configuración tiene sus propias características y limitaciones. En la descripción del protocolo SRTP se ha visto que éste ha sido diseñado para la

transferencia de medios de comunicación, sin embargo no es el adecuado para datos, por esta razón Datachannel está la parte superior de los protocolos UDP, DTLS y SCTP.

## 2.2.6. Establecimiento de Conectividad Interactiva

### 2.2.6.1. NAT

La función Network Address Translation (NAT) “se ha diseñado para simplificar y conservar direcciones IP. Permite que las inter redes IP privadas que usan direcciones IP no registradas puedan conectarse a Internet” (CISCO, 2008). Esta función se ejecuta por lo general en un enrutador, conectando dos redes, traduciendo las direcciones IP privadas de una red interna en direcciones legales antes de reenviar paquetes a otra red. NAT ofrece seguridad y conservación de red, pues generalmente se implementa en entornos de acceso remoto, reservándose el derecho de publicar todas las direcciones de una red al mundo exterior.



**Figura 1 El mundo real**

### 2.2.6.2. Interactive Connectivity Establishment (ICE)

ICE entra en acción, cuando WebRTC intenta establecer una conexión punto a punto entre dos usuarios. Como en el mundo real cada punto final está detrás de una o más capas de NAT o algunos de éstos tiene un software antivirus que bloquea ciertos puertos y protocolos, una dirección no puede simplemente intercambiarse con un compañero y comunicarse directamente.

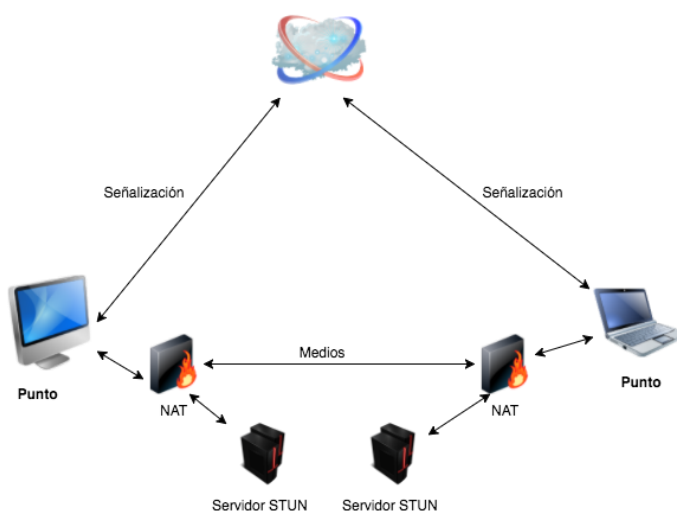
Para que se cree una conexión entre redes, la aplicación web debe pasar direcciones Uniform Resource Locator (URL) del servidor ICE a RTCPeerConnection. ICE trata de encontrar el camino más corto para conectar pares. Primeramente intenta



conectar utilizando direcciones de host obtenido del sistema operativo y la tarjeta de red, en el caso de que no funcione, es porque existe al menos un NAT por detrás, por lo pasará a obtener una dirección externa utilizando un servidor Session Transversal Utilities for NAT (STUN), pero si eso tampoco funciona, el tráfico se dirija a través de un servidor Transversal Usig Relays around NAT (TURN) (Dutton, 2013).

### 2.2.6.3. Session Transversal utilities for NATS (STUN)

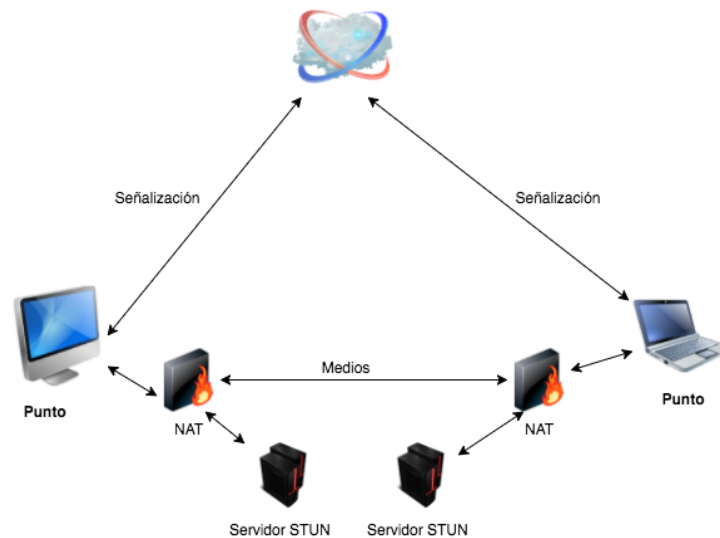
Un servidor STUN se aloja en el Internet público y se usa para obtener direcciones IP de acceso público (aplicaciones dentro de NAT) para que se pueda establecer una conexión entre pares (Dutton, 2013). ICE recurre a STUN cuando la comunicación directa de RTCPeerConnection mediante UDP ha fallado, recurriendo a TCP.



**Figura 2** Uso de STUN para obtener IPs públicas

### 2.2.6.4. Transversal Usig Relays around NAT (TURN)

Un servidor TURN apoya a STUN cuando la comunicación de RTCPeerConnection ha recurrido a TCP y ha fallado. Un servidor TURN es mucho más robusto y se usa para ser utilizado como punto de retorno, retransmitiendo segmentos de audio o video.



**Figura 3 Uso de Servidores STUN y TURN**

## 2.3. Servicio de videoconferencia múltiple

### 2.3.1. Stream

Una secuencia (Stream) es una abstracción de una secuencia de bytes, como un archivo, un dispositivo de entrada y salida, un conducto de comunicación entre procesos o un puerto de comunicaciones TCP/IP (Microsoft Developer, 2016). Con la aparición del Internet se ha facilitado la compartición de archivos, pero la transferencia de archivos de gran longitud, dificulta que estos puedan ser recibidos en tiempo real. Por esta razón el método ideal que se ha inventado con el afán de solucionar este problema es generar flujos de datos (Data Streams), para que a través del Internet, un cliente pueda reproducir los datos multimedia al mismo tiempo que recibe de un servidor.

### 2.3.2. Videoconferencia

Una videoconferencia es un servicio multimedia que permite la comunicación entre personas ubicadas geográficamente distantes, interconectando sesiones interactivas dependiendo del número de interlocutores, de tal forma que todos puedan compartir audio y video en tiempo real, para así simular una conversación como si estuvieran una sala de reuniones (Medina, 2003).

Este servicio facilita la comunicación bidireccional y simétrica, cada uno de los sitios de conexión se convierten en emisores y receptores al mismo tiempo (Medina, 2003). Sus principales características son la preparación de la señal digital, la

transmisión digital y el procesamiento de la señal digital recibida. Una vez que se ha digitalizado la señal, las transmisiones de video conferencia pueden ir sobre cualquier conexión alámbrica o inalámbrica.

Para que esta señal digital sea transmitida de forma óptima sobre alguna velocidad transmisión, se ha requerido de métodos de compresión que hagan que la longitud de la señal digital pueda ser lo más pequeña posible, sin la necesidad de perder la calidad del audio o video.

### **2.3.3. Códec**

Su nombre proviene de la abreviatura COdificador/ DECodificador, “es una implementación de software y hardware, capaz de transformar un archivo con un flujo de datos o una señal” (Cantos & Luna, 2007). Es un algoritmo o un método de compresión que se utiliza para codificar las señales audio o video en un formato de archivo.

Los codecs son de uso común en videoconferencias, donde la transmisión de imagen digital en movimiento requiere el almacenamiento y procesamiento de datos multimedia. Se estima el uso de un ancho de banda excesivo para una aplicación de este tipo, por ejemplo en (Carlos Genis Triana, 2004) se menciona que:

Para transmitir video crudo (sin codificar) en una resolución de 176x144, utilizando 24 bits de profundidad y una tasa de 10 frames/seg, se necesitaría un ancho de banda de casi 6 Mbps. Así, un segundo de un video de color con el estándar NTSC requiere casi 23 MB y un video de 90 minutos, en este mismo formato, requeriría aproximadamente 120 GB.

Dichos recursos no están disponibles para una aplicación de videoconferencia, por lo que es necesario el uso de los codecs de audio y video en aplicaciones de videoconferencia, puesto que se necesita reducir la cantidad de información que se genera, sobre todo en las imágenes con movimiento, analizando siempre la calidad de la imagen o el sonido no se vea afectado.

#### **2.3.3.1. Video en WebRTC**

WebRTC es capaz de enviar y recibir video interactivo, el cual puede provenir desde cualquier fuente, archivo, cámara, grabación de pantalla, etc. Para esta tarea se requiere del uso de uno de sus códec de video de los que provee el proyecto. Según

(IETF, 2016), no existe una normativa para la captura de la imagen, sin embargo a los beneficiarios del proyecto se les invita a tomar ventaja para obtener una mejor captura de los fotogramas. Estos trucos se basan en algunas configuraciones de la cámara, la pantalla y la orientación del lente.

Los codecs de video de la familia H.26X han evolucionado para las aplicaciones de telecomunicaciones a través del Sector de Normalización de las Telecomunicaciones de la UIT (UIT-T), sin embargo WebRTC trabaja también con otros codecs de la norma de codificación de Google VPX del proyecto WebM, los cuales brindan características diferentes a los de la familia antes mencionada.

### **H.263**

Es un estándar de UIT-T para codificación de video de baja velocidad binaria, desarrollada en 1995-1996 y diseñada para usar en sistemas basados en Public Switched Telephone Network (PSTN) en aplicaciones de videoconferencia y telefonía, aunque posteriormente se pudo utilizar en sistemas basados en los protocolos (RTP / IP) y sus versiones seguras (Amin, 2016).

### **H.264 / AVC**

Advanced Video Coding (AVC) es un estándar de UIT-T que se introdujo en 2003 para la codificación de video, capaz de satisfacer streaming de video a través de redes inalámbricas y móviles.

### **H.265 / HEVC**

High Efficiency Video Coding (HEVC) es la continuación de H.264 introducido en el 2013 para cumplir con la mejora de la resolución del video y utilizar el tratamiento en paralelo. HEVC soporta todas las aplicaciones de H.264 y tiene una sintaxis genérica que debe ser adecuado para las que no son H.264 (Amin, 2016). La estandarización de HEVC sólo incluye la sintaxis y la estructura de flujo de bits.

### **VP8**

En 2009 Google adquirió a On2 Technologies, la cual desarrollaba los codecs de video de la familia VPX. VP8 se lanza en el 2010 con el proyecto WebM con una visión

de poseer un códec de video abierto, algo no visto antes, y desde su lanzamiento sus investigadores se concentraron en analizar su rendimiento en las transmisión inalámbrica.

VP8 fue diseñado para aplicaciones web, por lo que requiere de un poco ancho de banda, también soporta la veracidad de los dispositivos de hardware que utilizan aplicaciones de contenido de video basado en la web y ejecutan eficazmente la compresión de video (Amin, 2016).

## **VP9**

Google comenzó el desarrollo del códec VP9 (sucesor de VP8) a finales del 2011 sobre el proyecto WebM. “Las herramientas de codificación en VP9 han madurado ya, y el flujo de bits se finalizó en junio de 2013” (Mukherjee & Bankoski, 2013). El desarrollo de VP9 nace con el objetivo explícito de crear un formato que produzca una forma mucho más compacta de flujo de bits que VP8, especialmente para el contenido de alta definición, pero con poco aumento en su complejidad de decodificación.

### **2.3.3.2. Audio en WebRTC**

## **OPUS**

OPUS es un códec de audio interactivo en tiempo real que fue lanzado en el 2012 e implementa la compresión de audio con pérdida. Se puede utilizar con tasas de bits altas y bajas (IETF, 2012). Éste códec fue diseñado para manejar una amplia gama de aplicaciones: llamadas punto a punto, conferencias, telepresencia, teleoperación y software remoto, juegos de chat de voz, etc (IETF, 2011), que eran requisitos para un códec de audio en Internet hasta esa fecha.

Una aplicación WebRTC que funciona con audio de alta calidad, la única opción de codecs de audio es OPUS, que trabaja con:

- Tasa de bits compatibles: constante y variable, de 6 kbit / s a 510 kbit/s.
- Tasas de muestreo compatibles: de 8 kHz a 48 kHz.

## **iSAC**

Internet Speech Audio Codec (iSAC) fue diseñado para aplicaciones de voice over IP (VoIP) y streaming de audio por Global IP Solutions, pero adquirido por Google en el 2011, quien lo hizo libre al incluirlo en el proyecto WebM. El códec iSAC es de banda ancha / superancha adaptativa que funciona con un retardo corto, por lo que es adecuado para la comunicación en tiempo real de alta calidad. Está especialmente diseñado para proporcionar una calidad de voz de banda ancha tanto para aplicaciones de baja y media velocidad de bits.

- Velocidades de bits compatibles: adaptación y variable. A partir de 10 kbit / s a 52 kbit / s.
- Tasas de muestreo compatibles: 32 kHz

## **iLBC**

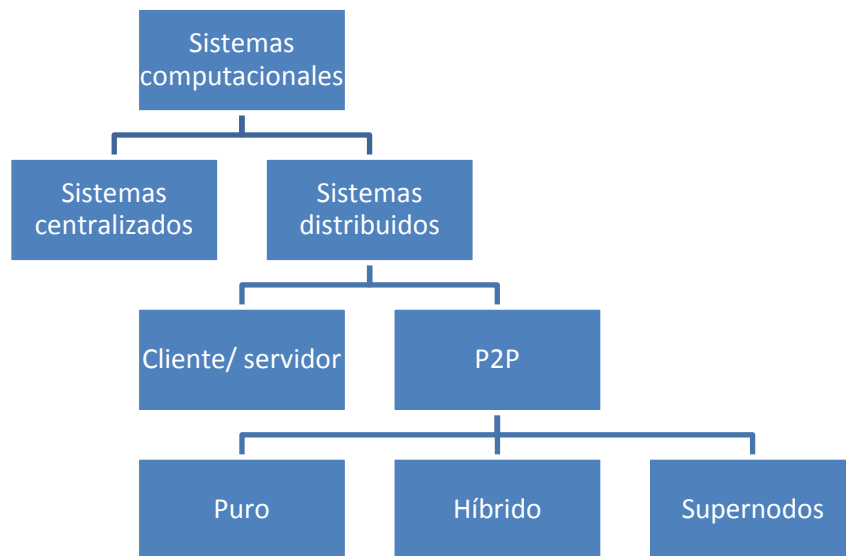
Internet Low Birate Codec (iLBC) fue lanzado en 2004 y también fue desarrollado por Global IP Solutions y adquirido luego por Google en el 2011. Es fuerte cuando la red tiene muy malos canales y bajo ancho de banda, pues trabaja con:

- Velocidades de bits compatibles: tasa de bits fija. 15,2 Kbit / s, o 13,33 Kbit / s.
- Frecuencia de muestreo: 8 kHz

## **2.4. Arquitectura del Servicio de Videoconferencia**

### **2.4.1. Peer to Peer (P2P)**

Punto a punto (P2P) es modelo de arquitectura de un sistema computacional distribuido. Se clasifica en tres grupos; el primero es un P2P puro, donde no existe un servidor central, el segundo es un híbrido, el cual contiene un servidor central para obtener información de los sensores de cada equipo a conectar, y finalmente está el de supernodos, el cual puede proveer de información a otros equipos de la red (Gatica, 2005).



**Figura 4 Clasificación de los sistemas de computación**

En la arquitectura Punto a punto, por su naturaleza distribuida y colaborativa, los costos de infraestructura son bajos y se puede alcanzar una alta escalabilidad (Amir, 2013). Las conexiones directas que poseen los puntos, proveen de mejores tiempos de respuesta entre ellos.

P2P hace referencia a un modelo de comunicación donde no existen clientes ni servidores fijos, todos los puntos se comportan al mismo nivel compartiendo a parte de los datos de información multimedia, datos específicos de cada equipo como: configuración local, velocidad de proceso, ancho de banda, capacidad de almacenamiento.

Los beneficios de esta comunicación se ven prácticamente en la reducción de costos en su infraestructura, visualización del estado actual de cada punto, pues cada uno utiliza sus propios recursos, y gracias al amplio uso de protocolos que usa, se puede utilizar en aplicaciones como: streaming de audio o video, mensajería instantánea, compartición de archivos.

#### **2.4.2. Protocolos de Comunicación**

Un protocolo de comunicación se compone de un grupo de reglas y formatos de mensajes establecidos de ante mano para que la comunicación entre sus partes sea posible. Las reglas definen la manera en que deben establecerse las comunicaciones de

las redes, como: temporización, secuencia, revisión y corrección de errores. Mientras que los formatos aseguran que sus mensajes sean válidos.

## **WebSocket**

Es una tecnología joven, aún en desarrollo, por lo que su implementación tiene algunos riesgos. Éste protocolo fue estandarizado por la (IETF, 2011), que define que: “El Protocolo WebSocket permite la comunicación bidireccional entre un cliente ejecutando código no confiable en un ambiente controlado a un host remoto que ha optado a las comunicaciones de ese código”.

Éste protocolo de la capa de aplicación está basado en el protocolo TCP independiente, su única relación con HTTP es la negociación que es interpretada por el servidor como una solicitud de actualización. Sobre la base de la recomendación de expertos de la Internet Assigned Numbers Authority (IANA), éste protocolo por defecto utiliza el puerto 80 para las conexiones regulares y el puerto 443 para las conexiones de túnel WebSocket sobre Transport Layer Security (TLS) (IETF, 2011).

Según (Mozilla, 2015) WebSocket es una tecnología avanzada que hace posible abrir una sesión de comunicación interactiva entre el navegador del usuario y un servidor, sin embargo sigue siendo una tecnología joven y sin implementar completamente en todos los navegadores, por lo que se pueden usar bibliotecas alternativas, como por ejemplo (socket.io, ws, jetty, pywebsocket, tornado, etc.), dependiendo del lenguaje de programación en el cual se vaya a desarrollar.

## **WSS**

El protocolo WebSocket es seguro si se utiliza WebSocket Secure (WSS) sobre los certificados SSL/TLS, donde la información es encriptada durante su transporte y se la protege contra los ataques de “el hombre en el medio”. La seguridad sobre el protocolo “ws://” es “wss://”, que indica una conexión segura sobre SSL/TLS.

## **HTTP**

El protocolo Hipertext Transfer Protocol (HTTP) de la capa de aplicación solo transmite texto, ha estado en uso por la World-Wide-Web desde 1990, desde la versión



“HTTP/ 1.0” (Berners-Lee, 1996). Las imágenes y archivos binarios se envían codificados en hexadecimal para cumplir con la característica del protocolo. No tiene estado, es decir no guarda ninguna información de conexiones anteriores.

## **HTTPS**

Los usuarios de Internet hoy en día dependen diariamente de HTTPS para la comunicación segura con sitios que tengan intención de visitar en su navegador web. Los servicios de seguridad, confidencialidad, integridad de mensajes y autenticación de HTTPS (HTTP sobre SSL / TLS) se fundamentan en la interacción correcta sobre las infraestructuras fuera de la banda, procedimientos y decisiones de confianza (Clark, 2013).

## **SSL**

El protocolo criptográfico Secure Sockets Layer (SSL) de la capa de transporte fue desarrollado para atender necesidades del navegador web Netscape para asegurar el tráfico web y específicamente diseñado para trabajar con HTTP (Rescorla, 2000). SSL se diseñó para proporcionar un canal general que puede ser adoptado con modificaciones mínimas en casi cualquier búsqueda basada en el protocolo TCP con algo de seguridad (Clark, 2013).

## **TLS**

El protocolo criptográfico Transport Layer Security (TLS) de la capa de transporte “actualiza el protocolo SSL público de más edad (v3.0). Proporciona un túnel seguro a un servidor, que es más comúnmente autenticado por un certificado X.509” (Clark, 2013). El objetivo de TLS es proporcionar privacidad e integridad de los datos entre dos aplicaciones que se comuniquen. Está compuesto por dos capas: Handshake y Record, la primera se basa en TCP para el registro TLS, mientras que la segunda tiene dos propiedades básicas (IEFT, 2008):

- Conexión privada, se usa la criptografía simétrica para el cifrado de datos, y éstos se generan únicamente para cada conexión y se basan en un secreto de negociación.

- Conexión fiable, el transporte de mensajes incluyen un mensaje de comprobación mediante una dirección Media Access Control address (MAC) con llave.

## **Certificación**

Un certificado web se compone de la unión de una clave de firma pública y una identidad, ésta composición se establece mediante una firma digital, donde el cliente acepta mantener la exactitud de la unión por un período de tiempo. TLS permite al software cliente establecer un canal confidencial determinado por la entidad que posea la clave privada asociada al certificado. El atributo esencial que todos los certificados de servidor HTTPS tienen es un nombre de dominio que los controla el titular del certificado (Clark, 2013).

## **TCP**

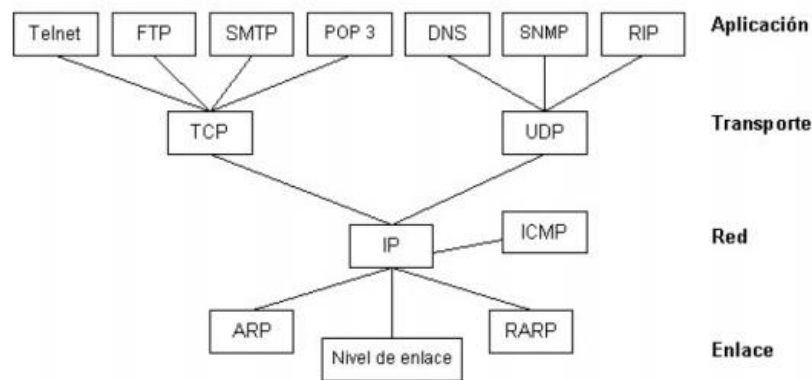
El protocolo de la capa de transporte Transmission Control Protocol (TCP) fue diseñado para funcionar de forma fiable durante cualquier medio de transmisión, independientemente de la tasa de transmisión, retardo, corrupción, duplicación o reordenación de segmentos (IEFT, 1992). TCP provee de una comunicación confiable, pero con una velocidad mínima para el uso aplicaciones de comunicaciones en tiempo real, ya que se asegura de añadir información extra en cada segmento para detectar y corregir en los datos transmitidos y recibidos. Lo que UDP solo tiene 8 bytes por segmento, TCP carga a 20 bytes.

## **UDP**

El protocolo de la capa de transporte User Datagram Protocol (UDP) es simple, de gran inteligencia que se comporta al mejor esfuerzo, no tiene información extra para la transferencia fiable de segmentos de datos, por lo que éstos se pueden perder o llegar desordenados, sin embargo existen aplicaciones que requieren más de velocidad que de fiabilidad, como lo es la transferencia multicast de audio y video. “UDP es usado por aplicaciones de VoIP, difusión de video y videoconferencia” (Candelas, 2009).

En realidad UDP no es mejor o peor que TCP, cada protocolo es mejor en cada uno de sus servicios para los que fueron diseñados. La (figura 5) siguiente distingue las

aplicaciones dedicadas para los protocolos de la capa de transporte, según la arquitectura de red TCP/IP.



**Figura 5 Conjunto de protocolos en la Arquitectura TCP/IP**

Fuente: (Candelas, 2009)

## SDP

El protocolo Session Description Protocol (SDP) es el “destinado a describir sesiones multimedia para los fines de anuncio de sesión, invitación de sesión, y otras formas de inicio de sesión multimedia” (IETF, 2006). Proporciona una representación estándar para el establecimiento de la comunicación entre pares, independientemente del protocolo por el cual se vayan a transportar los datos.

SDP es un protocolo de uso general y puede ser utilizado en varios entornos y aplicaciones de red. En WebRTC, la comunicación se negocia mediante mensajes de metadatos de SDP que describen las capacidades de los medios de comunicación (Google Developers, 2016).

WebRTC utiliza SDP para representar las capacidades del navegador web y preferencias basadas en texto, incluyendo la siguiente información (IETF, 2013):

- Capacidad de medios
- Audio
- Video
- Codecs utilizados

- Dirección IP y número de puerto
- Protocolo de Transmisión de Datos (Secure Real-time Transport Protocol SRTP)
- Ancho de banda utilizable

## **RTP**

El protocolo de la capa de Transporte Real-time Transport Protocol (RTP) está diseñado para transmitir medios de comunicación y es implementado en la parte superior de UDP. Los paquetes RTP en WebRTC incluyen un indicador de tiempo y un número de secuencia para mitigar los problemas de red como: retardos y paquetes fuera de orden (WebRTC Glossary, 2014).

## **RTCP**

El protocolo Real-time Transport Control Protocol (RTCP) se utiliza junto a RTP, pues lo provee de un mecanismo de control de peso ligero necesario para enviar informes estadísticos y mensajes de control de flujo.

## **DTLS**

WebRTC encripta la información usando datagramas de Transport Layer Security (DTLS), ofrece garantías de seguridad similar a TLS y se utiliza en WebRTC para el intercambio de clave de medio seguro en DTLS-SRTP. Todos los datos enviados a través RTCDataChannel se protegen utilizando DTLS (NTT Communications, 2015).

## **SRTP**

Secure Real-time Transport Protocol (SRTP) “es el único mecanismo de transporte utilizado por los medios de comunicación WebRTC” (WebRTC Glossary, 2014), se ubica en la parte superior de RTP, junto a SRTCP y provee de seguridad a RTP con las siguientes características:

- Integridad
- Autenticación

- Intimidad (tráfico cifrado)

## SCTP

SCTP (Stream Control Transmission Protocol) es un protocolo de transporte, similar a TCP y UDP, que se puede ejecutar directamente en la parte superior del protocolo IP. Sin embargo, en el caso de WebRTC, SCTP es tunelizado a través de un túnel seguro DTLS, que a su vez se ejecuta en la parte superior de UDP (Grigorik, 2013).

SCTP recoge las mejores características de TCP y UDP: de esta forma Datachannel API de WebRTC es orientada a mensajes, que provee un control de flujo, control de congestión, y configurable en los temas de confiabilidad y entrega de mensajes, como lo describe la siguiente tabla:

**Tabla 3**  
**Comparación de TCP, UDP y SCTP**

CRITERIO	TCP	UDP	SCTP
Confiabilidad	de confianza	sin confianza	Configurable
Entrega	ordenado	desordenada	configurable
Transmisión	byte orientado	mensaje orientado	mensaje orientado
Control de flujo	sí	no	sí
El control de congestión	sí	no	sí

Fuente: (Grigorik, 2013)

### 2.4.3. Patrón de Arquitectura Modelo Vista Controlador (MVC)

El patrón MVC fue escrito por primera vez en 1979, “surge con el objetivo de reducir esfuerzos en la programación de aplicaciones de sistemas múltiples y con los mismos datos” (Fernández & Díaz, 2012). Es un paradigma que divide la aplicación en tres entidades: el Modelo, la Vista y su Controlador, implementando por separado cada elemento para garantizar la actualización y el mantenimiento de software.

MVC se emplea en sistemas de representación gráfica, con ventanas separadas por módulos, lo cual permite una implementación por separado. Cualquier programador que use la API podrá modificar el modelo sin mucha dificultad, además que la conexión entre el modelo y sus vistas es dinámica, pues se produce en tiempo de ejecución y no de compilación (Fernández & Díaz, 2012).

- Modelo: maneja los datos y controla sus transformaciones, no tiene referencia a los controladores o las vistas.
- Vista: maneja la presentación visual de los datos presentados en el modelo, de preferencia interactúa con el controlador, aunque también puede tratar con el modelo a través de una referencia.
- Controlador: entra en función cuando se realiza algún cambio, pues es quien interpreta las órdenes del usuario e interactúa con el modelo y la vista.

#### **2.4.4. Patrón de diseño de estructura Proxy**

Una aplicación que provea movilidad debe estar diseñada con un agente o servicio proxy para gestionar transparentemente los cambios de conectividad (Yoro, 2006). Consiste en interponer un intermediario entre un objeto y los demás que lo utilizan, manteniendo una referencia al objeto real y controlando el acceso a éste. Existen diferentes tipos de proxy:

- Proxy remoto.- codifica peticiones y envían al objeto real.
- Proxy virtual.- almacena información del objeto real
- Proxys de protección.- comprueba permisos de los clientes para realizar una petición.

## **2.5. Estudios de Calidad**

### **2.5.1. Calidad de Servicio (QoS)**

La calidad de servicio (QoS) es un conjunto de tecnologías que proporcionan la capacidad de administrar el tráfico de red de manera rentable y mejorar las experiencias de usuario. En (Murazzo, Rodríguez, Vergara, Carrizo, González, & Grosso, 2013)se menciona que:

El soporte de QoS se ha convertido en un factor importante dentro de los servicios convergentes actuales. De esta manera es posible trabajar sobre una plataforma de red cableada, inalámbrica o móvil, dándole soporte adecuado a servicios emergentes, como voz, video o datos.

En una red de comunicación del servicio de videoconferencia, se requiere de una elevada calidad de voz y video de un punto a otro. Se describen a continuación algunos de los factores más importantes que afectan a la calidad del servicio:

### Jitter

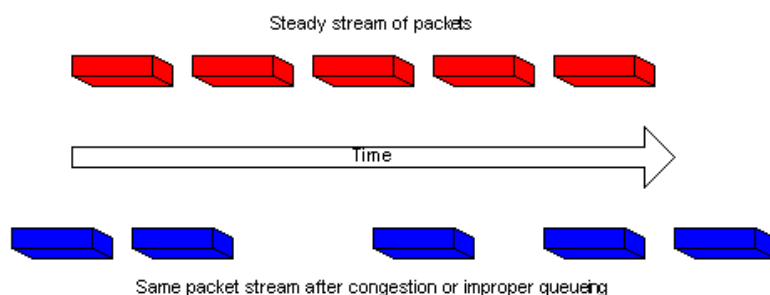
“El jitter se define técnicamente como la variación en el tiempo en la llegada de los paquetes, causada por congestión de red, pérdida de sincronización o por las diferentes rutas seguidas por los paquetes para llegar a su destino” (CISCO, 2006). Se calcula al tomar la diferencia en milisegundos de los tiempos de retraso por dos paquetes secuenciales indexados como  $k$  y  $k + 1$ , donde  $\tau_{k,d}$  es la cola de retraso para el paquete indexado  $k$ , valor que puede ser negativo, pero por temas prácticos se toma su valor absoluto (Hammad, 2016).

$$\Delta t(k + 1, k) = \tau_{k+1,d} - \tau_{k,d}$$

#### Ecuación 1 Cálculo del Jitter

Fuente: (Hammad, 2016)

En las redes basadas en conmutación de paquetes, el jitter es un efecto que se da cuando un paquete toma un distinto camino de los otros paquetes para llegar a su destino, por la razón de que las redes IP no definen una trayectoria específica para el transporte de paquetes de datos. En consecuencia, los paquetes pueden llegar a su destino en un nivel de sincronización diferente al que han sido enviados, o en el peor de los casos, pueden ser eliminados.



**Figura 6 Manejo de un flujo de paquetes**

Fuente: (CISCO, 2006)

## Retraso

Una de las características más importantes en el servicio de videoconferencia es la temporalidad, esto significa que tanto como el audio como el video debe de aparentar ser presentado en tiempo real al receptor. “El retraso es el tiempo tomado por un paquete en alcanzar el punto final de recepción después de ser transmitido desde un punto de envío” (Murazzo, Rodríguez, Vergara, Carrizo, González, & Grosso, 2013).

El retraso se calcula al tomar la diferencia en milisegundos del tiempo de recepción del paquete anterior  $T_{n-1}$  y el tiempo de llegada del paquete  $T_n$  que por casos prácticos se toma su valor absoluto (Marcillo, 2015). Minimizar ésta diferencia es muy importante, por ejemplo: en el caso de la voz humana, si una palabra de dos sílabas se pronuncia con un retraso entre la una y la otra, se pierde la melodía de la voz, causando inteligibilidad del mensaje y molestias en la calidad del servicio y experiencia del usuario.

$$\text{Retraso} = T_{n-1} - T_n$$

### **Ecuación 2 Retraso**

Fuente: (Marcillo, 2015)

## Eco

Se refiere a la reflexión de la señal en el otro extremo, la ITU ha establecido un umbral de 5 mili segundos, donde el emisor, se convierte en receptor de sus propias palabras. A medida que el retardo se incrementa, la posibilidad de establecer una conversación disminuye notablemente, además de presenciar un gran descontento por parte del usuario.

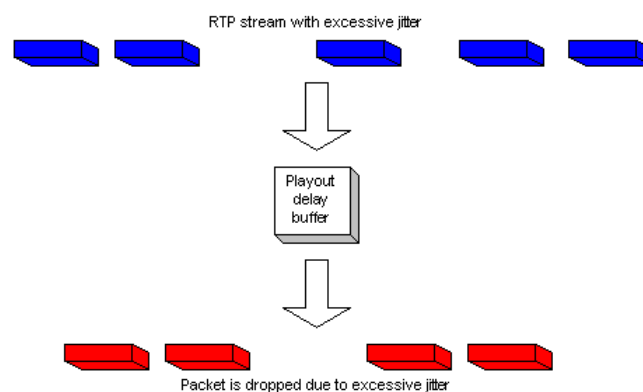
## Solapamiento de la voz de los interlocutores

Son las pausas que se dan en la transmisión de voz, las cuales invitan a contestar al otro interlocutor cuando este aún no haya recibido el mensaje por completo, el umbral de este retardo es aceptable hasta los 150 mili segundos según (Huidrobo & Roldan, 2003).



## Pérdida de paquetes y fuera de orden de video

Para evitar este problema se ha necesitado de la implementación de algoritmos de reordenación de paquetes a partir de los tiempos que se incluyen en los paquetes. Los paquetes se pueden perder debido a congestión de la red, y a las técnicas de manejo de la congestión. Además, si un paquete desordenado tarda más en llegar que la duración del buffer de reproducción del paquete, se le puede contabilizar como un paquete perdido. (Díaz, Merino, Panizo, & Recio).



**Figura 7 Manejo de un excesivo jitter**

Fuente: (CISCO, 2006)

### 2.5.2. Calidad de Experiencia (QoE)

Calidad de experiencia (QoE) es el grado de aceptabilidad de una aplicación o servicio como una percepción subjetiva del usuario final (ITU International Telecommunication Union, 2007). El estudio de esta calidad incluyen también los efectos de inicio a fin del sistema, como son: cliente, terminal, red, servicios de infraestructura, etc. Además de las expectativas del usuario y el contexto en que se desenvuelve.

Por los años 1990 ya se hablaba sobre el concepto de Calidad de Servicio (QoS), el mismo que tuvo una gran atención en las telecomunicaciones y posteriormente se desencadenó en varios estudios sobre el estado del servicio respecto a la opinión del usuario (Tavakoli, 2015). Y desde ese entonces se ha adoptado la definición de la ITU con las características de ser un estudio subjetivo centrado en el usuario.

Sin embargo, la más reciente definición de QoE fue publicada por (Patrick Le Callet, 2013) a través de una publicación en una revista indexada, con la finalidad de avanzar con la convergencia en cuanto a este tema, diciendo que:

QoE es el grado de placer o disgusto del usuario de una aplicación o servicio. Es el resultado de la realización de sus expectativas con respecto a la utilidad y/o el disfrute de la aplicación o servicio en función de la personalidad del usuario y el estado actual.

Esta última definición aporta con la importancia del estudio de la utilidad y disfrute, que son aspectos relacionados en una valoración de calidad. Por otro lado, según la tesis de (Tavakoli, 2015), las características de QoE se clasifica en 4 niveles: percepción directa, situación de uso, el servicio e interacción. Pero debido a que la comunidad científica no ha llegado a una definición común de QoE, con un marco unificado, la definición de la UIT sigue siendo la más aplicada.

Para medir “calidad de experiencia de video a través de redes inalámbricas, hay tres enfoques de evaluación diferentes, las cuales son, subjetiva, objetiva e híbrida” (Amin, 2016), las cuales se describen a continuación:

- **Pruebas Objetivas:** Es un método indirecto para medir la calidad de experiencia, puesto que el resultado del estudio de la calidad de audio y video se correlaciona con el promedio obtenido de la observación de los evaluadores (pruebas subjetivas). La Aplicación de estas pruebas es cara, éste enfoque es usado por investigadores que están buscando técnicas de evaluación alternativas que empleen fórmulas y algoritmos para recopilar datos de los parámetros técnicos. Éste enfoque tiene varias métricas de evaluación diferentes (Amin, 2016), donde la más común es (Peak Signal-to-Noise Ratio PSNR).
- **Pruebas Subjetivas:** Es un método directo por el cual se puede medir la calidad de experiencia, en donde la observación de un grupo de evaluadores es el resultado final.
  - **Métrica de evaluación directa Mean Opinion Score (MOS):** es un resultado de pruebas subjetivas de los evaluadores. El mismo expresa un

juicio de calidad promedio de un grupo, al cual se le ha asignado una misma condición de prueba. Su resultado se basa en una escala original de cinco niveles (Excelente, Bueno, Justo, Pobre, Malo) (Tavakoli, 2015).

**Tabla 4**  
**Mean Opinion Score MOS**

MOS	Calidad	Discapacidad
5	Excelente	Imperceptible
4	Buena	Perceptible pero no molesta
3	Regular	Ligeramente molesta
2	Pobre	Molesta
1	Mala	Muy molesta

Fuente: Trad. Del Inglés (Amin, 2016).

- **Pruebas Híbridas:** Para proporcionar una medición más precisa QoE, los investigadores crearon una evaluación híbrido que combina la evaluación subjetiva y parámetros objetivos, y esta evaluación se llama Pseudo Subjective Quality Assessment (PSQA) (Amin, 2016).

## 2.6. TRABAJOS RELACIONADOS

Las investigaciones sobre mecanismos de control de desconexiones de redes inalámbricas en los servicios de transmisión de información en tiempo real de consumo son relativamente nuevas en tecnologías de implementación libre, más aún sobre el framework WebRTC. Con el fin de mitigar los retrasos y pérdidas de información en servicios de streaming se han adoptado algunas alternativas: la implementación de sistemas adaptativos, el uso de agentes inteligentes y algoritmos de reconexión de la red son las mejores propuestas y han sido la base de ésta investigación.

En (Skirelis & Serackis, 2015) con la finalidad de brindar un mejor servicio de streaming de video se propone mitigar el jitter, retardo y la pérdida de los paquetes enviados a través de la conexión. Propone el uso de la implementación de un sistema inteligente basado en un algoritmo en datos estadísticos de WebRTC, de esta manera le provee de alto rendimiento a la aplicación ya que el sistema tiene la capacidad de decidir en tiempo real sobre la calidad de video que puede rendir a conexión.

En la tesis doctoral de (Marcillo, 2015) se presentan tres soluciones que se pueden implementar en el servicio de servicio de streaming de video móvil sobre dispositivos

Android que minimice la pérdida de paquetes enviados, pues trabaja sobre la potencia de la señal inalámbrica, el retardo y su variación. La primera solución se basa en un modelo basado en patrones de diseño y arquitectura de software, el cual mitiga las interrupciones de la sesión con el uso de agentes inteligentes implementados sobre JADE-Android. Donde el uso de MVC permite la integración del patrón Proxy para orquestar e intermediar el mecanismo de control de interrupciones entre el cliente y el servidor. La segunda solución es complementaria a la primera y proporciona de realidad aumentada a la interacción con el usuario, utiliza un censado de la intensidad de señal inalámbrica RSSI para presentar sobre un mapa de cobertura local con la ayuda del lenguaje de marcado de posiciones geográficas Keyhole Markup Language (KML) como lo describe la (figura 8).



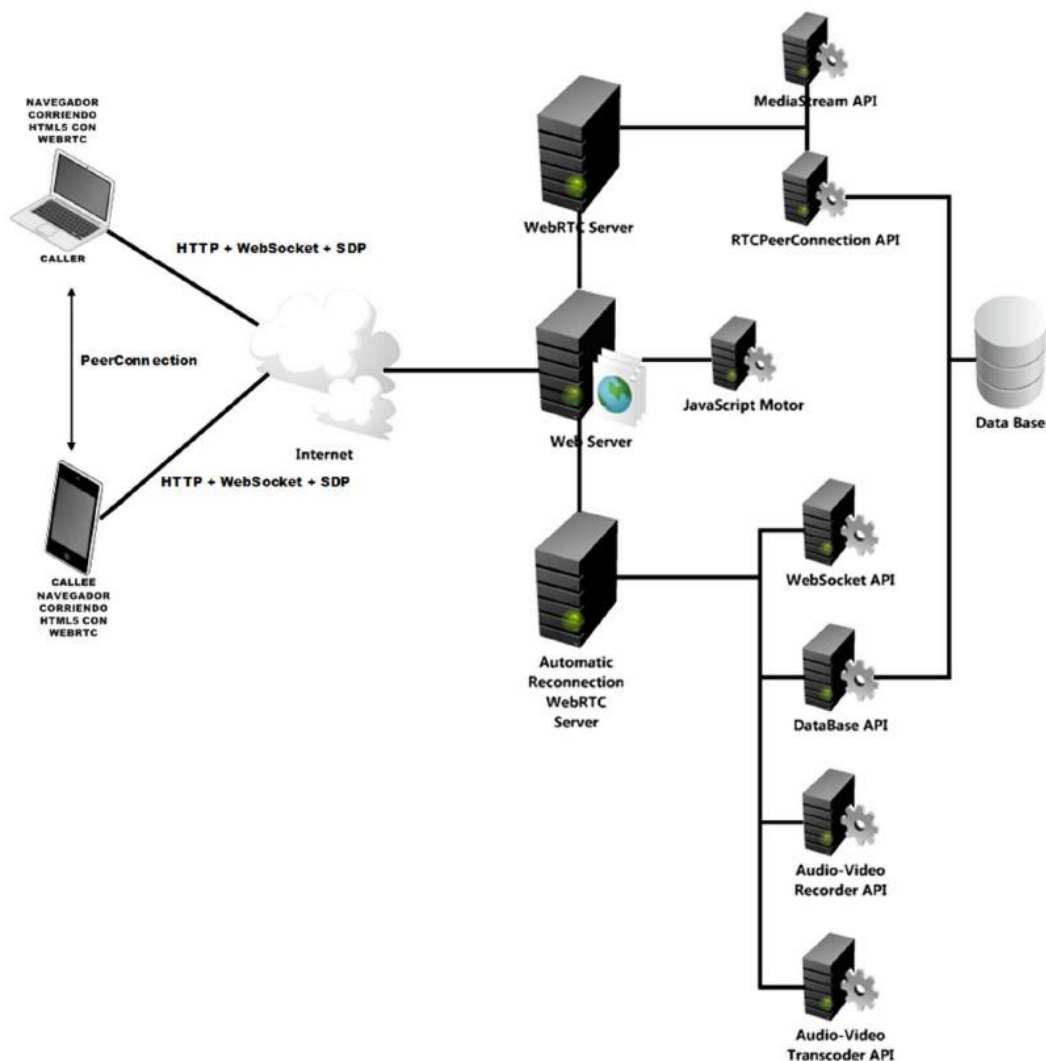
**Figura 8 Arquitectura con Realidad Aumentada**

Fuente: Trad. Del Inglés (Marcillo, 2015)

Y la más acertada y última solución que en conjunto con la tesis de (Yacchirema & Ortega, 2015) liberan la solución a un ambiente multiplataforma gracias al desarrollo de un mecanismo de control de desconexión que mitiga la pérdida de datos en una video llamada P2P mediante la implementación de un algoritmo de reconexiones de sesión que le permitiera mantener su estado. Esta solución se desarrolló sobre el framework WebRTC, utilizando los patrones MVC y Proxy, el orquestador del establecimiento de la reconexión del canal de comunicación se dio gracias a la librería de Socket.io de NodeJS.

La mitigación de la pérdida de la información se basa en una grabación de multimedia en el navegador web, la (figura 9) describe como la información era

almacenada en un servidor central, el cual enviaba a los puntos en cuanto se reconectarán. Lamentablemente la problemática de esta solución es que no brindaba la posibilidad de la interacción de más de dos usuarios concurrentes, limitando su alcance, además de almacenar en disco toda la información grabada, haciendo al sistema inseguro, además que su transmisión de datos no era cifrada.



**Figura 9 Arquitectura Automatic Reconnection WebRTC (ARW)**

Fuente: (Marcillo, 2015)

Por otro lado, (Kaimov, 2010) propone otro enfoque a la solución de la problemática de las desconexiones de sesión en el servicio de streaming de video, su trabajo utiliza un gestor en la parte de la construcción de la red P2P basada en árboles para proporcionar para volver a conectar. La ventaja es que se puede monitorizar pares y se proporciona información en tiempo real sobre sus pares de la red.

Los enfoques existentes en la actualidad pueden dividirse en tres clases principales: los enfoques basados en los árboles, los enfoques basados en la malla y los enfoques basados en el árbol de malla híbridos. Mientras que el primer enfoque es ideal para la multidifusión de video en (Kaimov, 2010), el enfoque basado en malla ha de ser el que soporte el servicio de videoconferencia múltiple, donde varios usuarios se comunican entre sí, siendo servidores y receptores de varios canales de comunicación al mismo tiempo. Esto no quita que se pueda utilizar el enfoque de árbol de malla híbrido o supernodos para establecer un mejor servicio y obtener varias salas de videoconferencia múltiple.

## CAPÍTULO III

### DISEÑO, DESARROLLO, IMPLEMENTACIÓN, IMPLANTACIÓN

#### 3.1. Introducción

El presente capítulo presenta en dos partes el desarrollo de la investigación, en una primera sección se describe la necesidad y la construcción de una plataforma de servicio de videoconferencia múltiple. Mientras que la segunda se presenta el desarrollo del mecanismo de control de desconexión y su implantación sobre el servicio base.

La descripción de cada una de las partes conlleva una introducción, análisis, diseño, pruebas funcionales y control de iteraciones de las historias de usuario. Todo bajo los criterios del uso de la metodología XP y el uso del patrón de arquitectura MVC. Aquí no se presentan pruebas de calidad, éstas requieren un análisis matemático más profundo, por lo que ésta sección solo responde al cumplimiento de las historias de usuario.

#### 3.2. Componentes elementales de una videoconferencia

Un Sistema de videoconferencia genérico debe contener al menos los siguientes componentes por cada participante debido a que la comunicación se realiza en forma bidireccional.

Se requiere de una entrada de video y audio que permitan capturar tanto la imagen como la voz de los participantes con el fin de convertirlas en una señal digital durante el procesamiento para posteriormente ser transmitida. Además es necesaria una unidad de procesamiento que se encargue de convertir la información obtenida de las entradas de audio y video en un flujo de datos para transmitirse, y de recibir la información y convertirla en un formato presentable en el receptor. La información viaja a través de un medio de transmisión, como el Internet o las redes telefónicas, que conecte y permita la comunicación entre los participantes. Finalmente la información recibida es presentada en un formato aceptable para el participante receptor a través de dispositivos de salida. (Thakur, 2012)

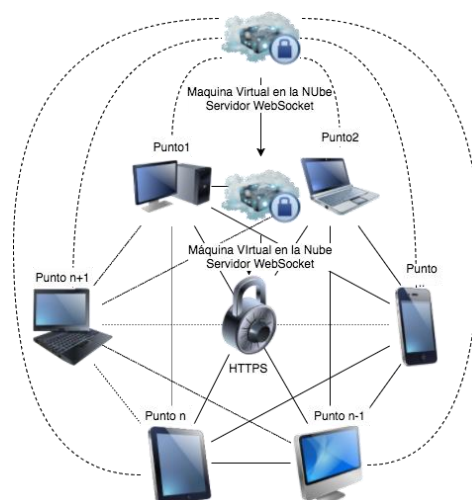
### 3.3. Plataforma base del servicio de videoconferencia en la nube

El desarrollo de una plataforma base del servicio de videoconferencia se ha visto necesario incluirlo como parte del proyecto de investigación, puesto que es importante poseer una aplicación construida sobre la tecnología WebRTC, de libre modificación y que incluya los requisitos fundamentales para cumplir con el fin del proyecto.

Además es importante usar los mismos patrones tanto como para la plataforma base como para el mecanismo de control de desconexión, con el objetivo de reducir esfuerzos para los investigadores. Pero antes que todo hay que aclarar qué tipo de sistema se ha de desarrollar. Como de ante mano se conoce que WebRTC maneja una conexión de navegadores como iguales, entonces el sistema ha de ser distribuido punto a punto. Sin embargo existen tres tipos de modelos de P2P; puro, híbrido y supernodos.

En un principio se pensaría que se usaría un P2P puro, pero se ha visto la necesidad de incluir un servidor en la nube que tenga la tarea de establecer las comunicaciones de los puntos, entonces se opta con un sistema distribuido P2P híbrido como lo describe la (figura 10). No se toma el modelo de supernodos porque no se necesita que un punto sea el servidor central de otra red, éste modelo se podría escoger en el caso de que se desarrollara una plataforma en donde un usuario pueda participar en varias salas de videoconferencia.

#### Sistema distribuido P2P



**Figura 10 Sistema Distribuido P2P híbrido**



En versiones anteriores de herramientas de Google sobre su navegador Chrome el usuario permitía el uso de la cámara del dispositivo aceptando los riesgos de seguridad a los que se exponía al usar un protocolo no seguro como lo es HTTP. En el lanzamiento de la versión 47 del navegador en Diciembre del 2015 se anuncia que se requiere permitir una sola vez el acceso a la cámara para HTTPS o el host local (Google Developers, 2016). Por esta razón el servicio de videoconferencia múltiple para n usuarios con diferentes plataformas y/o sistemas operativos, requiere de un certificado SSL/TLS para poder ofrecer la seguridad en todo el sistema mediante HTTPS.

### **3.4. Análisis y diseño de la arquitectura basado en patrones**

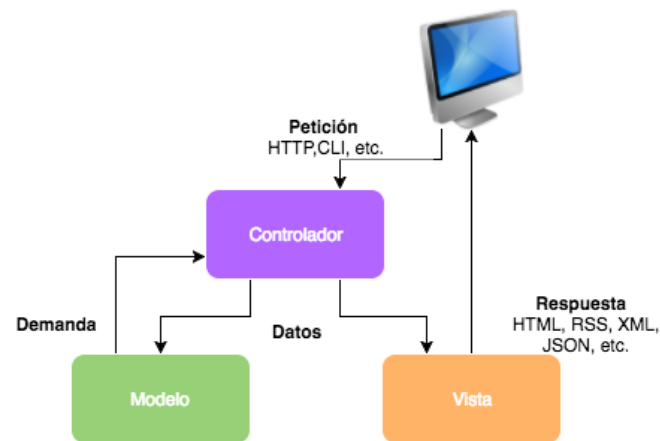
Se propone el uso del patrón Modelo-Vista-Controlador MVC con el objetivo principal de facilitar la reutilización de la arquitectura de software. Por principio un patrón brinda una solución a una problemática particular que se repite muchas veces y como (Castejón, 2004) dice “uno de los patrones que ha demostrado ser fundamental a la hora de diseñar páginas web es MVC”, en este proyecto se lo ha tomado en cuenta por las siguientes razones:

- Propone en la separación de la interfaz de usuario del modelo de negocio y el control.
- Las páginas web son páginas HTML, que serían visualizadas en un navegador.
- El usuario interactúa con el controlador de la aplicación mediante peticiones HTTP.

Sin embargo se tiene que dar una separación de la vista con el modelo, y para solventar esta problemática, el modelo ha de ser un servidor, el cual tenga el acceso a la base de datos de los usuarios de la plataforma de videoconferencia, cumpliendo completamente con las características del patrón.

La (figura 11) ilustra la interacción de las entidades del modelo de una aplicación web, donde el controlador es el encargado de recibir las peticiones HTTP a través de la interacción del usuario con la aplicación y demandar datos procesados del modelo, el

cual tiene acceso a los registros de las base de datos. Además éste controlador con sus reglas de gestión trabaja con los datos recibidos del modelo para enviarlos a la vista, la cual mediante algún lenguaje de marcas (HTML), son mostradas en el navegador web.



**Figura 11 Arquitectura MVC**

La Arquitectura propuesta es descrita por la (figura 12), el modelo y el controlador corren sobre una máquina virtual en la nube, la misma que posee el sistema operativo Linux con Python 3 y servidor Apache. El servidor WebSocket cuenta con dos módulos, el modelo accede a una base de datos Mysql y el controlador de eventos que actúa como un proxy remoto, pues abstrae peticiones y valida el estado de conexión para establecer la video llamada entre los puntos.

El controlador también es formado por el módulo de los eventos WebRTC, ubicado en cada punto (máquina cliente), se encarga de censar la información los puntos, en especial del estado de la conexión de ellos con la ayuda de la API RTCPeerConnection. Finalmente, el usuario podría interactuar con la plataforma usando un navegador Google Chrome y visualizar el contenido que provee el módulo de la vista compuesto de páginas HTML, con etiquetas en su versión 5, especialmente <audio> y <video> con ayuda del framework Bootstrap para su diseño adaptativo.

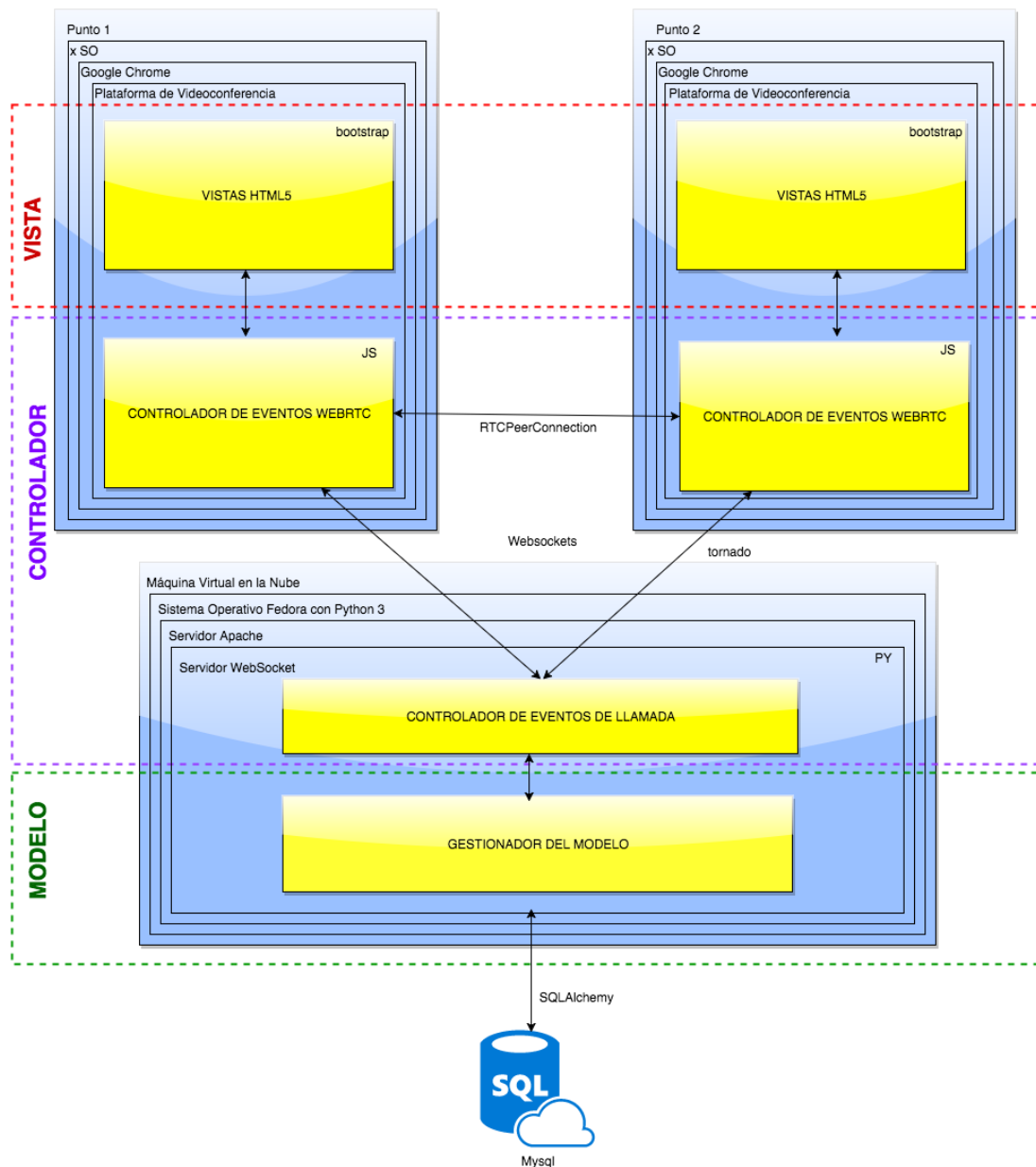


Figura 12 Arquitectura MVC de Videoconferencia

### 3.5. Metodología XP en la plataforma base de videoconferencia

#### 3.5.1. Historias de Usuario

Las funcionalidades de la plataforma base se basa en las siguientes historias de usuario, las cuales definen el número de iteraciones a ser controladas en el desarrollo de software.

**Tabla 5**  
**Historia de usuario - Ingreso al sistema**

Historia de Usuario	
Número: 1	Usuario:
Nombre historia: Ingreso al sistema	
Prioridad en negocio: Alta (Alta/Media/Baja)	Riesgo en Desarrollo: Bajo (Alto, Medio, Bajo)
Puntos Estimados:	Iteración asignada: 1
Programador responsable: Damián Luje – Raúl Llasag	
Descripción: Tener un registro que diferencie a cada usuario y además lo provea de un clave privada para el ingreso seguro al sistema.	
Observaciones: se debe poder cerrar la sesión.	

**Tabla 6**  
**Historia de usuario - Contactar usuario**

Historia de Usuario	
Número: 2	Usuario:
Nombre historia: Contactar Usuario	
Prioridad en negocio: Alta (Alta/Media/Baja)	Riesgo en Desarrollo: Medio (Alto, Medio, Bajo)
Puntos Estimados:	Iteración asignada: 2
Programador responsable: Damián Luje – Raúl Llasag	
Descripción: Enviar invitaciones de contacto a otros usuarios y recibir invitaciones que puedan ser aceptadas o rechazadas.	
Observaciones: Si un usuario acepta una invitación de contacto, su identificación debe ser encontrada dentro de una lista de contactos.	

**Tabla 7**  
**Historia de usuario - Sala de videoconferencia común**

Historia de Usuario	
Número: 3	Usuario:
Nombre historia: Sala de videoconferencia múltiple	
Prioridad en negocio: Alta (Alta/Media/Baja)	Riesgo en Desarrollo: Alta (Alto, Medio, Bajo)
Puntos Estimados:	Iteración asignada: 3
Programador responsable: Damián Luje – Raúl Llasag	
Descripción: Llamar a un contacto para agregarlo a una sala de videoconferencia común para n usuarios.	
Observaciones: Un usuario puede rechazar su participación al terminar la llamada o negar su inicio.	

### 3.5.2. Casos de Uso

La plataforma base del servicio de videoconferencia consta de tres amplios casos de uso por la necesidad de cumplir con una llamada que involucre a varios usuarios en una misma sala de conferencia. Para ello se ha visto la necesidad de un registro de usuarios con la finalidad de que sean identificados al momento de ingresar a la plataforma, así cada contacto podrá buscar, solicitar y aceptar, para finalmente agregarlo a una sala de videoconferencia con dos o más usuarios (videoconferencia múltiple).

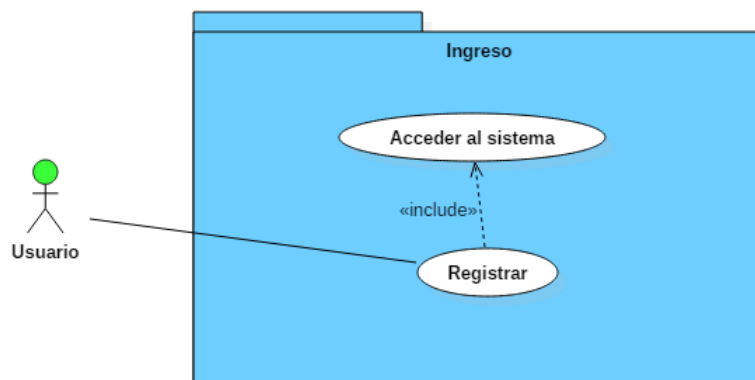


Figura 13 Caso de uso - Ingreso a la plataforma

Tabla 8

#### Descripción del caso de uso - Ingreso

<b>Caso de Uso: Ingreso</b>
<b>Actores:</b> Usuario
<b>Resumen:</b> el usuario podrá crear una cuenta mediante el registro e ingresar al sistema utilizando dicha cuenta.
<b>Precondiciones:</b>
<b>Descripción:</b> Registro: 1. Ingresar la información (Nombre, ID, Contraseña) 2. Aceptar Acceso al sistema: 3. Ingresar el Id y la contraseña 4. Aceptar
<b>Post condiciones:</b>
<b>Observaciones:</b> el usuario puede cerrar sesión.

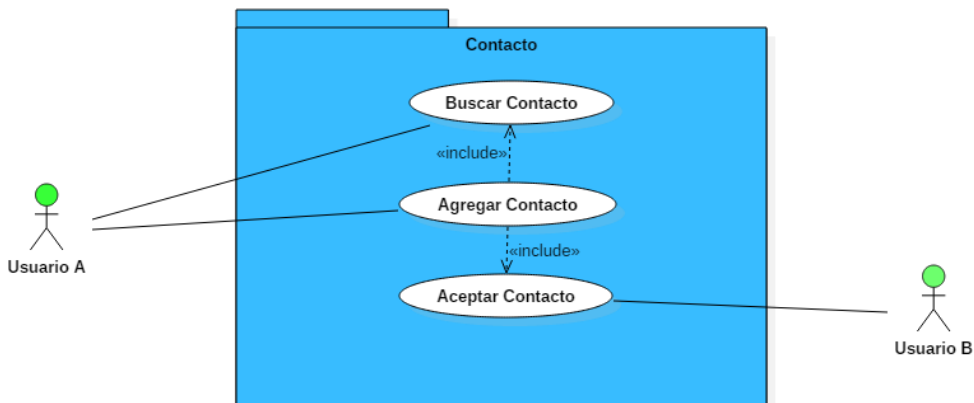


Figura 14 Caso de uso - Contacto

Tabla 9

Descripción del caso de uso - Contacto

<b>Caso de Uso: Contacto</b>
<b>Actores:</b> Usuario A y Usuario B
<b>Resumen:</b> un usuario (Usuario A) podrá agregar a otro usuario (Usuario B) como contacto, y el Usuario B podrá aceptar al Usuario A
<b>Precondiciones:</b> los usuarios deben haber ingresado al sistema.
<b>Descripción:</b> Agregar contacto: 1. Buscar el usuario 2. Enviar invitación Aceptar contacto: 1. Aceptar la invitación
<b>Post condiciones:</b> los usuarios serán establecidos como contactos y podrán realizar llamadas
<b>Observaciones:</b>

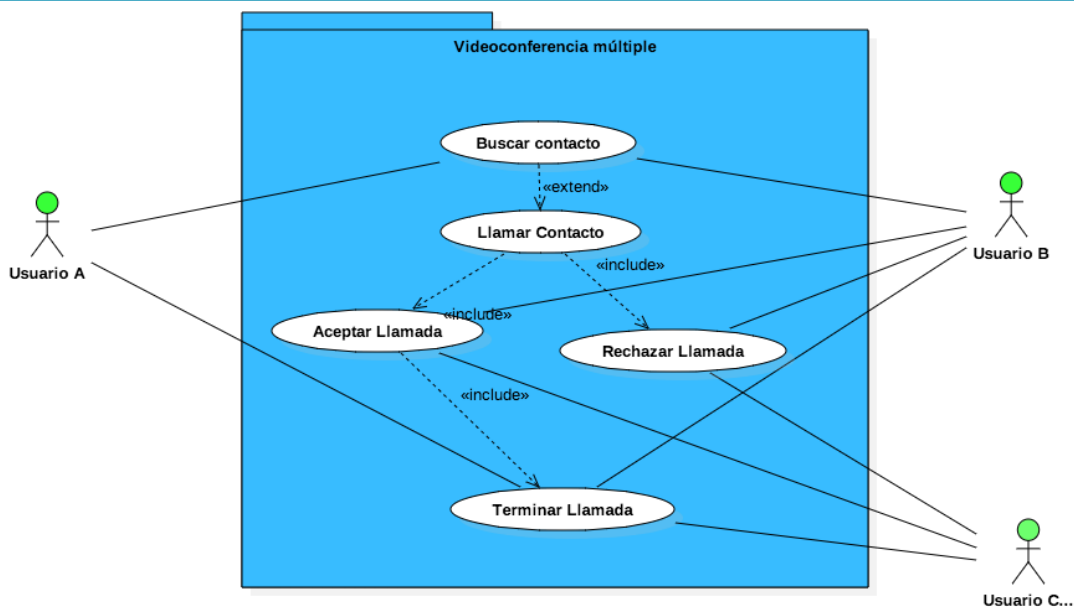


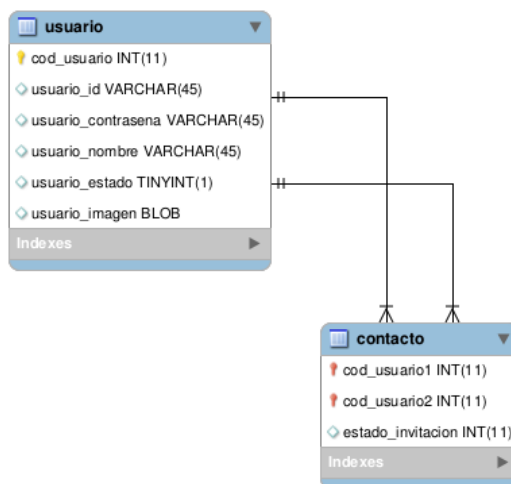
Figura 15 Caso de uso - Videoconferencia múltiple

**Tabla 10**

**Descripción del caso de uso - Videoconferencia múltiple**

<b>Caso de Uso: Videoconferencia múltiple</b>
<b>Actores:</b> Usuarios
<b>Resumen:</b> un usuario podrá llamar a otro usuario que se encuentre en línea, y cualquier usuario en la llamada puede agregar a otro
<b>Precondiciones:</b> todos los usuario deben haber ingresado al sistema
<b>Descripción:</b> Buscar contacto: 1. Ingresar el Id del usuario requerido 2. El sistema buscará al dicho usuario entre los contactos Llamar contacto: 3. Llamar al contacto encontrado 4. El sistema notificará al usuario a quien se realiza la llamada. Aceptar o rechazar llamada: 5. El usuario a quien se realizó la llamada podrá aceptar o rechazar la llamada Terminar Llamada: 6. Cualquier usuario de la llamada puede salir de la misma, terminando su comunicación con todos los otros usuarios de la llamada.
<b>Post condiciones:</b>
<b>Observaciones:</b> <ul style="list-style-type: none"> <li>• Un usuario en una llamada podrá agregar a cualquiera de sus contactos a la llamada.</li> <li>• Si un usuario termina la llamada y existen aún otros usuarios en dicha llamada, solo se eliminará al usuario que terminó la llamada y se mantendrá la comunicación entre los demás usuarios.</li> </ul>

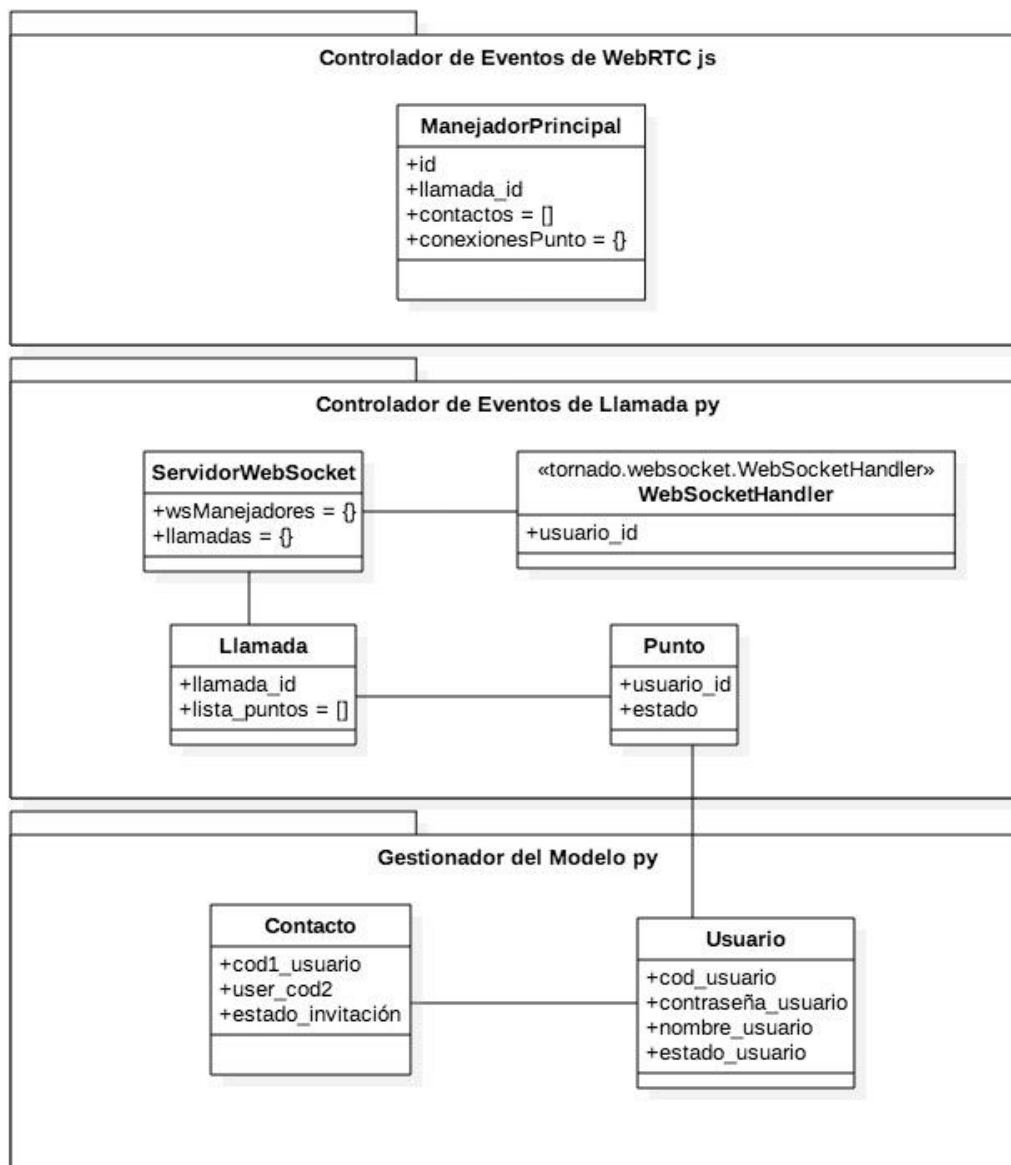
**3.5.3. Modelo Conceptual de Bases de Datos**



**Figura 16 Modelo de BD**

**3.5.4. Diagrama de Clases**

El diagrama de clases de la (figura 17) responde al diagrama de la arquitectura MVC propuesta por la (figura 12), el modelo y el controlador están construidos sobre dos lenguajes de programación diferentes: Python v3 (py) y JavaScript (js).

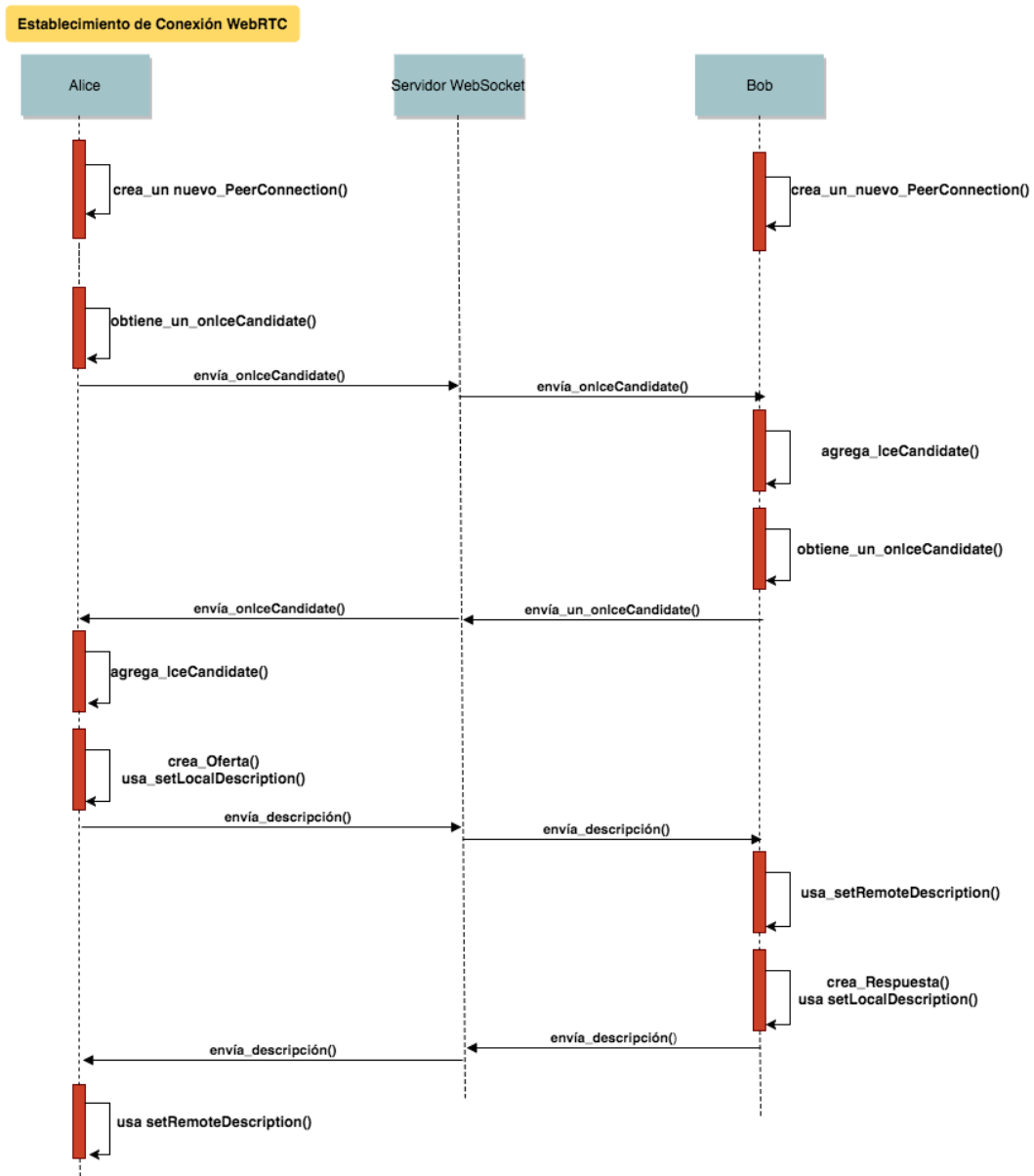


**Figura 17 Diagrama de Clases de videoconferencia**

### 3.5.5. Diagramas de Secuencia

Los tres diagramas de secuencia que se presentan en esta sección son la base de la interacción de los objetos y sus acciones esenciales para obtener un servicio de videoconferencia en la nube. El primero de ellos es la (figura 18), describe el establecimiento de la conexión punto a punto mediante la API `RTCPeerConnection` de WebRTC. Para este ejemplo a los puntos se les ha asignado los nombres ficticios de Alice y Bob, donde el primer punto llama al segundo y el diagrama de secuencia termina cuando ya establecen una comunicación al agregar una descripción remota.



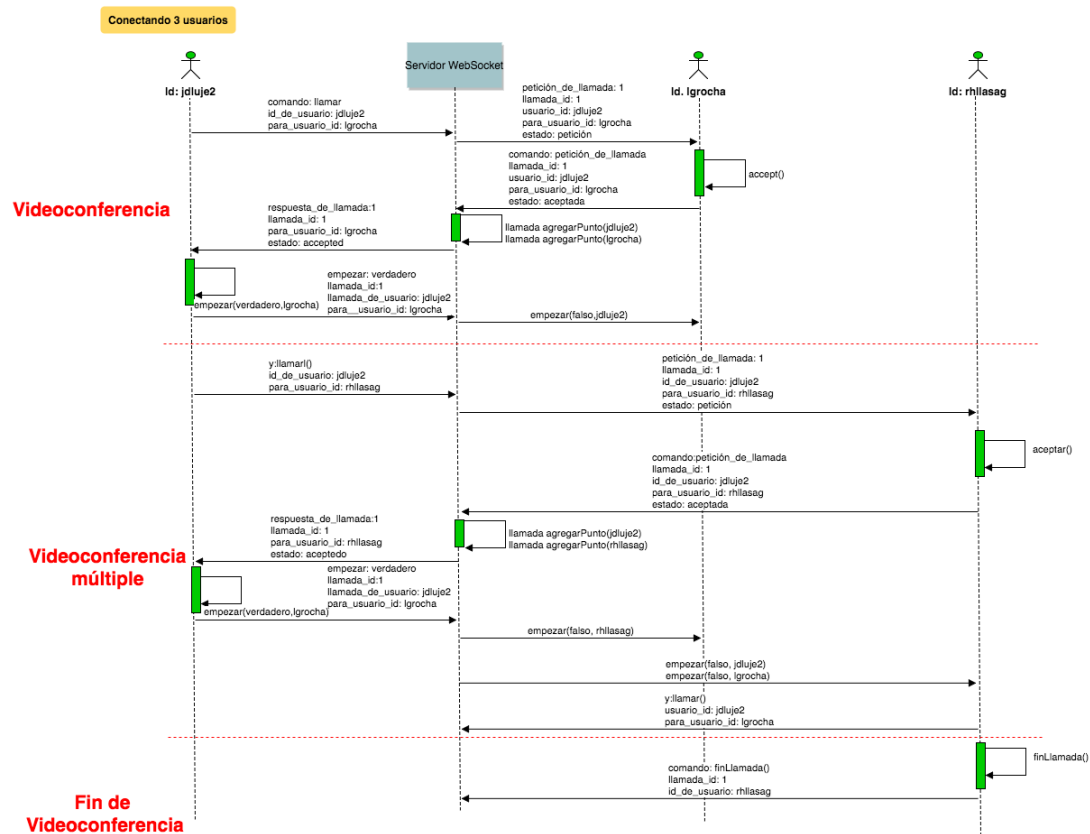


**Figura 18 Alice llama a Bob**

Se observa que existe un Servidor WebSocket, su importancia viene descrita desde una perspectiva a alto nivel con la arquitectura de la aplicación. Es el encargado de establecer la conexión de los puntos por el momento, pero en la segunda entrega de esta investigación, cuando se necesite también restablecer una comunicación de una videoconferencia se le deben incorporar funcionalidades.

El segundo diagrama, representado por la (figura 19) describe el establecimiento de la conexión de tres usuarios en una misma sala de conferencia. Para el manejo del ciclo de vida de la videoconferencia son necesarios algunos indicadores que describan;

el estado, los usuarios involucrados, la llamada a la que se integran, si desean integrarse o rechazan su participación.



**Figura 19 Ciclo de vida de una videoconferencia con tres usuarios**

Finalmente, el último diagrama de secuencia, representado por la (figura 20), describe la interacción planeada de los usuarios con el sistema. Cuando interactúan más de dos usuarios para comunicarse entre sí en una misma sala se llama videoconferencia, y aunque el gráfico solo contiene 3, este modelo se puede extender a  $n$  usuarios.

Diagrama de secuencia videoconferencia múltiple

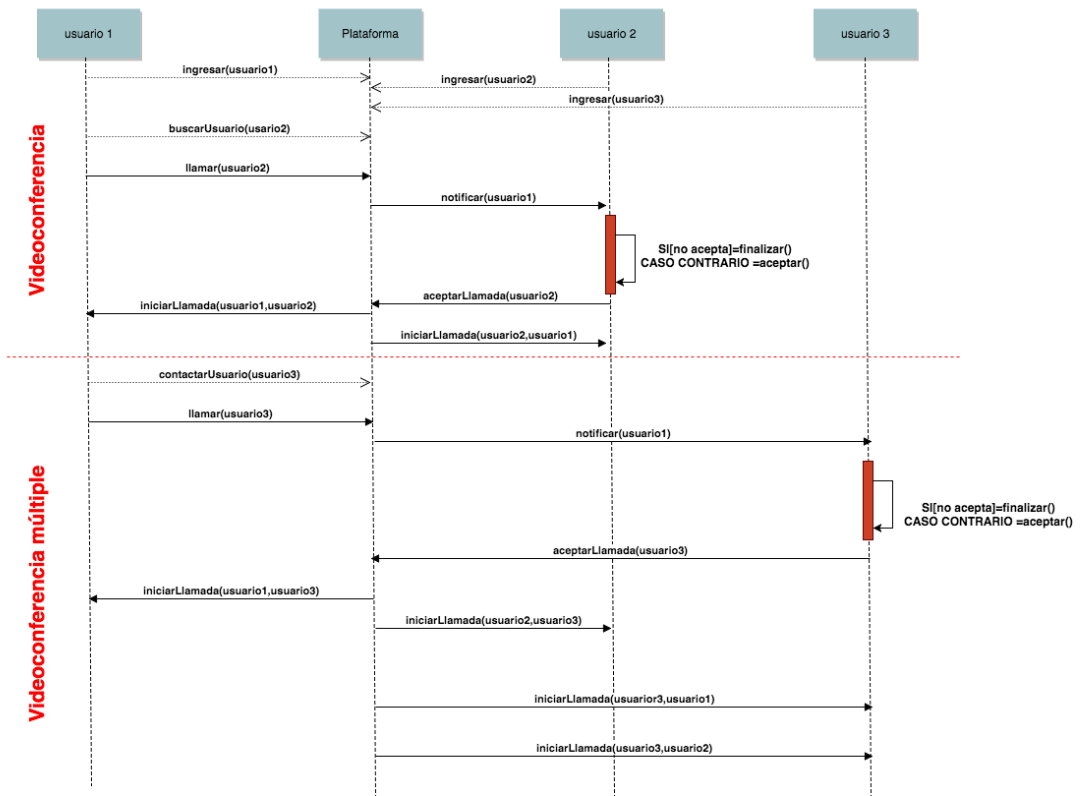


Figura 20 Diagrama de secuencia de videoconferencia múltiple

### 3.5.6. Control de iteraciones

**Tabla 11**  
**Control de iteraciones del desarrollo de videoconferencia múltiple**

HU	Tarea	Actividad	Descripción	Desarrollador	Fecha de inicio	Fecha final
1	Ingreso al sistema	<ul style="list-style-type: none"> <li>• Proveer una base de datos Mysql para registro de usuarios.</li> <li>• Crear un módulo de ingreso a la plataforma, validando registros.</li> <li>• Proveer de seguridad de la clave del usuario con cifrado md5.</li> <li>• Gestión de sesiones</li> </ul>	<ul style="list-style-type: none"> <li>• Inicio de sesión mediante la lectura de un usuario registrado.</li> <li>• Ocultación visual de claves de usuario.</li> <li>• Apertura y cierre de sesiones en el navegador mediante cookies.</li> </ul>	Raúl Llasag Damián Lujé	2016-03-15	2016-04-04
2	Contactar usuario	<ul style="list-style-type: none"> <li>• Gestión de invitaciones de contacto.</li> </ul>	<ul style="list-style-type: none"> <li>• Invitar, aceptar, rechazar invitaciones de contactos.</li> <li>• Buscar un usuario de entre la lista de contactos.</li> </ul>	Raúl Llasag Damián Lujé	2016-03-29	2016-04-05
3	Sala de videoconferencia múltiple	<ul style="list-style-type: none"> <li>• Brindar el servicio de videoconferencia múltiple</li> </ul>	<ul style="list-style-type: none"> <li>• Establecer una video llamada entre dos usuarios.</li> <li>• Integrar n usuarios a una misma sala de video llamada.</li> <li>• Finalizar llamada</li> </ul>	Raúl Llasag Damián Lujé	2016-04-05	2016-05-25

## 3.5.7.Pruebas Funcionales

Tabla 12

## Caso de prueba - Registro de usuario

<b>Campos a diligenciar</b>	Nombre ID Contraseña
<b>ID Campo de Prueba</b>	PF001
<b>Nombre caso de prueba</b>	Registro de usuario
<b>Descripción</b>	El usuario llena sus datos para registrarse
<b>Precondiciones</b>	
<b>Relaciones Caso de Uso</b>	Ingreso
<b>Pasos y condiciones de ejecución</b>	<ol style="list-style-type: none"> <li>1. Un usuario escribe su nombre de usuario, su id y contraseña</li> <li>2. Se registra en la plataforma.</li> </ol>
<b>Resultado esperado</b>	Registro en la base de datos
<b>Estado caso de prueba</b>	Exitoso
<b>Resultado obtenido</b>	Se ejecuta de acuerdo a lo planeado
<b>Errores asociados</b>	ID antes registrado
<b>Responsable diseño</b>	Raúl Llasag
<b>Responsable ejecución</b>	Damián Lujé

Tabla 13

## Caso de prueba - Ingreso al sistema

<b>Campos a diligenciar</b>	ID Contraseña
<b>ID Campo de Prueba</b>	PF002
<b>Nombre caso de prueba</b>	Ingreso a la plataforma
<b>Descripción</b>	Un usuario escribe su ID de usuario y su contraseña para ingresar a la plataforma.
<b>Precondiciones</b>	Registro de usuario
<b>Relaciones Caso de Uso</b>	Ingreso
<b>Pasos y condiciones de ejecución</b>	Llenar campos de ID de usuario y contraseña Pulsar el botón de ingresar.
<b>Resultado esperado</b>	Visualización de la pantalla principal
<b>Estado caso de prueba</b>	Exitoso
<b>Resultado obtenido</b>	Se ejecuta de acuerdo a lo planeado
<b>Errores asociados</b>	ID o contraseña no válidos
<b>Responsable diseño</b>	Raúl Llasag
<b>Responsable ejecución</b>	Damián Lujé

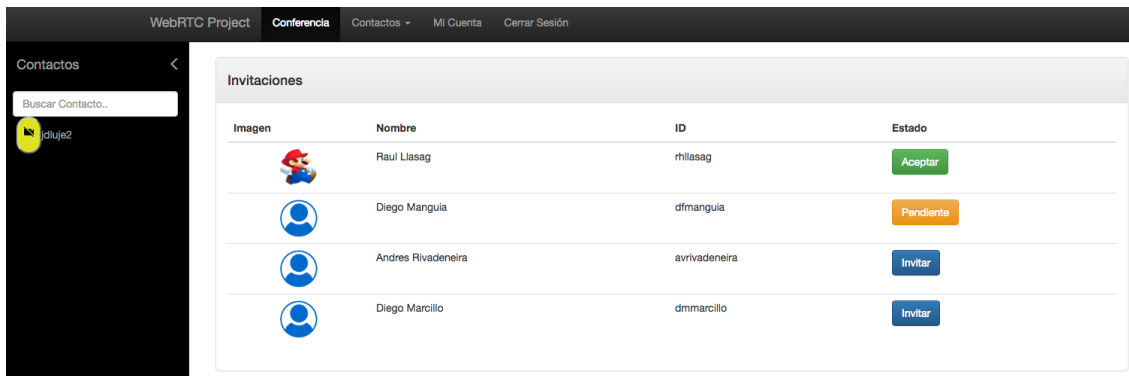


Figura 21 Caso de prueba – Contactar usuario 1

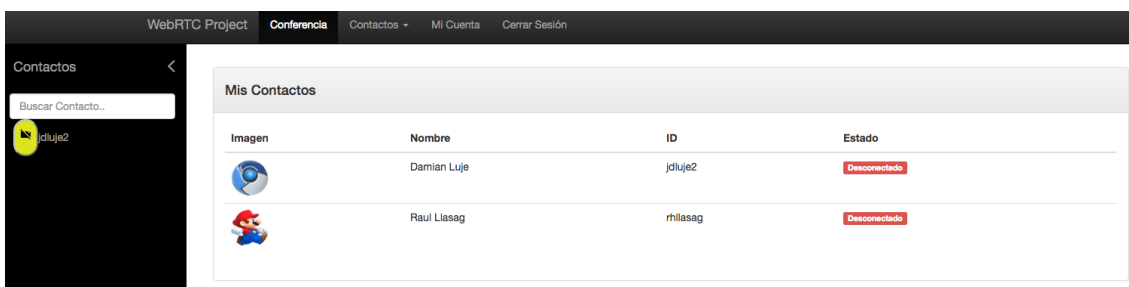


Figura 22 Caso de prueba – Contactar usuario 2

Tabla 14

Caso de prueba - Contactar usuario

<b>Campos a diligenciar</b>	Contacto
<b>ID Campo de Prueba</b>	PF003
<b>Nombre caso de prueba</b>	Contactar usuario
<b>Descripción</b>	Un usuario acepta una solicitud de contacto de otro usuario
<b>Precondiciones</b>	Registro a la plataforma por parte de los usuarios implicados
<b>Relaciones Caso de Uso</b>	Contacto
<b>Pasos y condiciones de ejecución</b>	<ol style="list-style-type: none"> <li>1. Iniciar sesión con el usuario solicitante.</li> <li>2. Ir a Contactos/ Invitaciones.</li> <li>3. El usuario solicitante escoge de una lista de usuarios a uno y lo invita a ser su contacto.</li> <li>4. Cerrar sesión</li> <li>5. Iniciar sesión con el usuario solicitado.</li> <li>6. Ir a Contactos/ Invitaciones.</li> <li>7. El usuario solicitado acepta la invitación de una lista de solicitudes de contacto.</li> <li>8. Ir a Contactos/ Mis contactos</li> </ol>
<b>Resultado esperado</b>	El nombre de usuario se visualiza en la lista de contactos.
<b>Estado caso de prueba</b>	Exitoso
<b>Resultado obtenido</b>	Se ejecuta de acuerdo a lo planeado
<b>Errores asociados</b>	
<b>Responsable diseño</b>	Raúl Llasag
<b>Responsable ejecución</b>	Damián Lujé

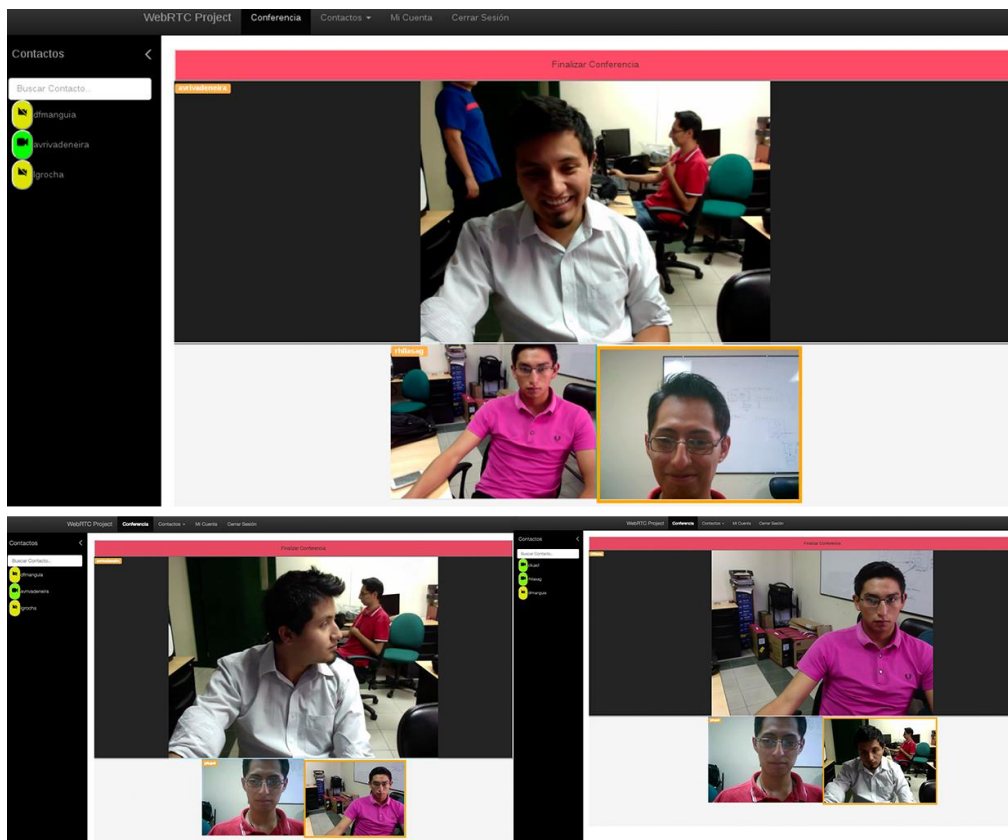


Figura 23 Caso de prueba – Sala de videoconferencia común

Tabla 15  
Caso de prueba - Sala de videoconferencia común

<b>Campos a diligenciar</b>	Audio y video en videoconferencia
<b>ID Campo de Prueba</b>	PF004
<b>Nombre caso de prueba</b>	Sala de videoconferencia común
<b>Descripción</b>	El usuario solicitado acepta una llamada de otro usuario.
<b>Precondiciones</b>	Ingreso a la plataforma por parte de los usuarios interlocutores.
<b>Relaciones Caso de Uso</b>	Videoconferencia múltiple
<b>Pasos y condiciones de ejecución</b>	<ol style="list-style-type: none"> <li>1. El usuario solicitante busca de entre una lista de contactos que el usuario solicitado se encuentre en línea para posteriormente poderlo llamar.</li> <li>2. El usuario solicitado acepta la llamada.</li> </ol>
<b>Resultado esperado</b>	Se visualiza en pantalla la imagen de los usuarios implicados en esa sala de videoconferencia y su id, así como también se escuchan sus voces.
<b>Estado caso de prueba</b>	Exitoso
<b>Resultado obtenido</b>	Se ejecuta de acuerdo a lo planeado
<b>Errores asociados</b>	Un usuario puede desconectarse de la red.
<b>Responsable diseño</b>	Raúl Llasag
<b>Responsable ejecución</b>	Damián Lujé

### **3.6. Problemas a causa de interrupciones del canal de comunicación**

En una videoconferencia en tiempo real sobre la web, si uno de los interlocutores pierde la señal de Internet, los otros pierden contacto y dejan de recibir su información, a pesar de que el emisor sigue enviando datos. Éstos, en los servicios de streaming se convierten en paquetes perdidos y su cantidad es directamente proporcional al tiempo de desconexión

Según (Marcillo, 2015), en sus estudios menciona que “complejo es el hecho que el canal inalámbrico tiene un comportamiento impredecible ocasionado por la desconexión impredecible del canal inalámbrico que causa interrupciones del servicio de video”, el mismo que puede afectar a la plataforma de videoconferencia con pérdidas de sesión sobre sus canales de comunicación.

Se han experimentando estos mismos comportamientos sobre la plataforma base de videoconferencia múltiple, obteniendo pérdidas de información al presenciar interrupciones de sesión y en la conexión entre pares. Este comportamiento trasciende en un lapso de tiempo de desconexión, en el cual toda la información que debería transmitirse, termina por perderse y no se recupera de alguna forma.

Sin embargo, esta no es la única problemática, existen también pérdidas de información a causa de la demora de un usuario al iniciar una nueva llamada luego de presenciar una desconexión, razón por la cual se requiere de una reconexión automática del canal de comunicación, que sea rápida y transparente para sus interlocutores.

Por otro lado, existen problemas que no se pueden controlar en un gran porcentaje, por ejemplo: en (Yacchirema & Ortega, 2015) realizan una reconexión y recuperación de información de una videollamada con dos interlocutores, enfocando su estudio en el estado de conexión de WebRTC, donde el tiempo en que se demora un punto en transcodificar el audio y video que envían al servidor y posteriormente recuperarlo en un formato legible. Pero en el presente estudio se trata de minimizar mucho más la pérdida, al no requerir del servidor para transcodificar el audio y video, siendo esta una tarea por el navegador de cada punto.



### **3.7. Mecanismo de control de desconexión**

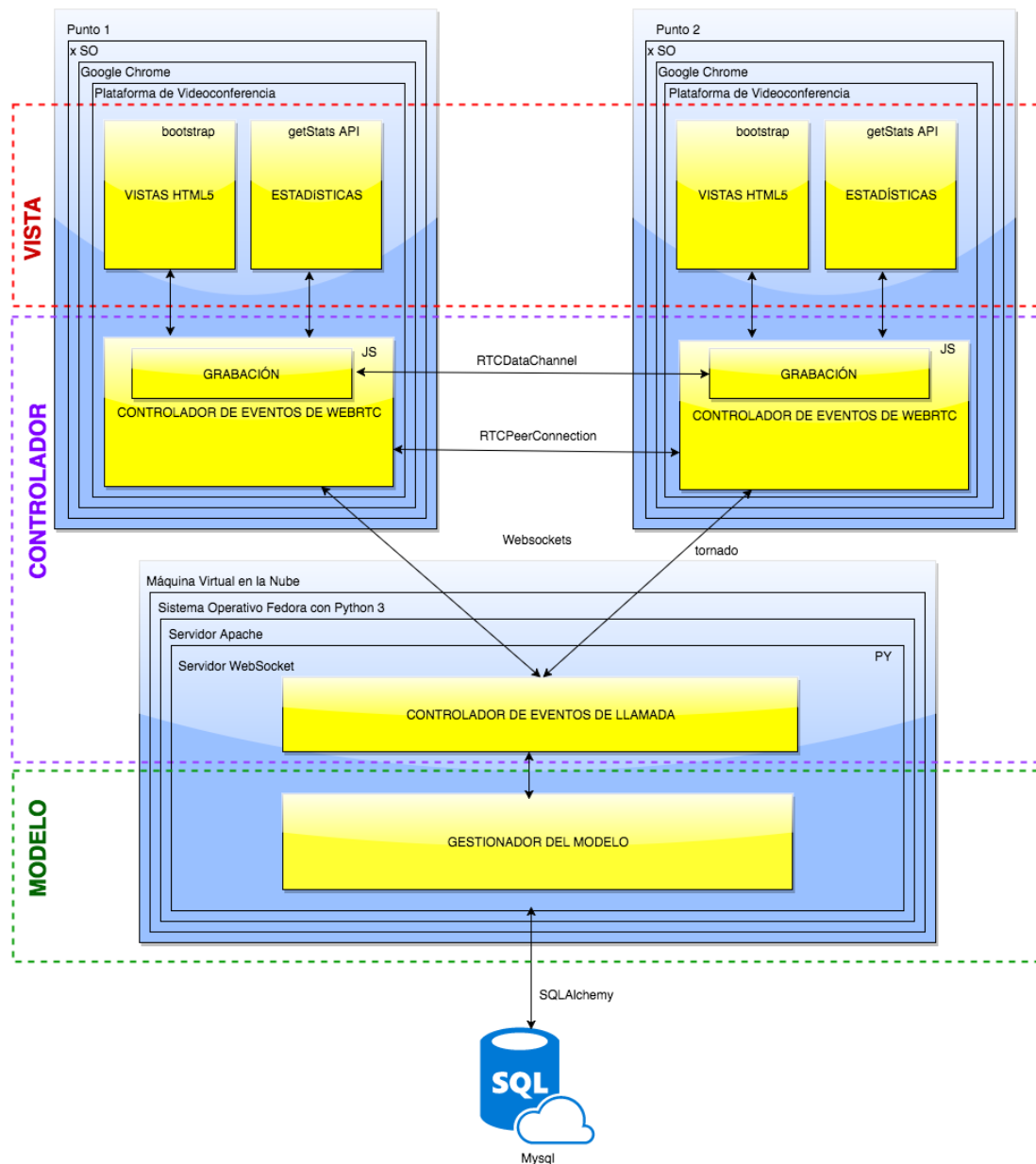
El desarrollo de éste mecanismo sigue la misma metodología XP y usa las mismas herramientas que la plataforma del servicio de videoconferencia múltiple implementado en la anterior parte. Sin embargo, a pesar de que pudo haber sido añadida su construcción como una iteración más, el mecanismo de control de desconexión se ha tomado como un proyecto separado porque conceptualmente puede ser implantado a otros servicios de comunicaciones en tiempo real sobre la web, razón por la que requiere mayor explicación desde su diseño.

### **3.8. Análisis y Diseño del mecanismo basado en patrones**

WebRTC trabaja con P2P, aquí los equipos involucrados (puntos) en la transferencia de información difieren en el estado de la conexión de la red. La API RTCPeerConnection provee de información de la conexión de una videoconferencia, pero no la puede establecer, con la ayuda de la librería tornado de websockets de Python los puntos pueden establecer y reestablecer la conexión.

A la arquitectura del servicio de videoconferencia múltiple definida por la (figura 12) se le ha añadido elementos para cumplir con el objetivo de controlar las desconexiones y la mitigación de la pérdida de la información. En (figura 24) se evidencia que se ha agregado un canal extra de comunicación entre los puntos, éste es el encargado de intercambiar información almacenada en disco de las grabaciones de la cámara y el micrófono mediante el códec de video ligero para la web VP9 en el módulo de grabación. Cuando un punto se ha desconectado y reconectado la transmisión de video se logra gracias a la API DataChannel de WebRTC.

Para brindar una conectividad continua de una sala de videoconferencia múltiple se necesita extender las funcionalidades del controlador de eventos de llamadas. “Una aplicación diseñada para movilidad debe trabajar con un agente o servicio proxy” (Yoro, 2006), razón por la cual el módulo controlador de eventos de llamadas codifica las peticiones de cada punto, verifica el estado de conexión y establece la comunicación con los demás puntos.



**Figura 24 Arquitectura del mecanismo de control de desconexión**

Por otro lado, y adelantándose a la fase de pruebas se ha añadido también un módulo de estadísticas a la capa de la vista, con la finalidad de obtener una variación en el tiempo del jitter, retardo, paquetes y ancho de banda utilizado a lo largo del ciclo de vida de toda la llamada. Pese, a que existe un módulo llamado webrtc-internals, éste no provee información separada por cada conexión, forma no óptima para esta investigación, por esto se ha recurrido al uso de getStats API y la implementación de un módulo de estadísticas propio.

Esta arquitectura además de proveer de reconexión al servicio de videoconferencia múltiple, también tiene la capacidad de proveer a otros servicios que mantengan el mismo modelo de sistema distribuido P2P híbrido, como por ejemplo streaming de video y compartición de archivos.

### 3.9. Metodología XP en el mecanismo de control de desconexión

#### 3.9.1. Historias de Usuario

Las funcionalidades del mecanismo de control de desconexión se basan en las siguientes historias de usuario, las cuales definen el número de iteraciones a ser controladas en el desarrollo de software.

**Tabla 16**

#### Historia de usuario - Reconectar llamada

Historia de Usuario	
Número: 1	Usuario:
Nombre historia: Reconectar llamada	
Prioridad en negocio: Alta (Alta/Media/Baja)	Riesgo en Desarrollo: Alto (Alto, Medio, Bajo)
Puntos Estimados:	Iteración asignada: 1
Programador responsable: Damián Luje – Raúl Llasag	
Descripción: Si un usuario presenta desconexión de red, reconectar el canal de comunicación en el momento en que vuelva a la red.	
Observaciones: reconectar solo si el usuario no ha terminado la llamada.	

**Tabla 17**

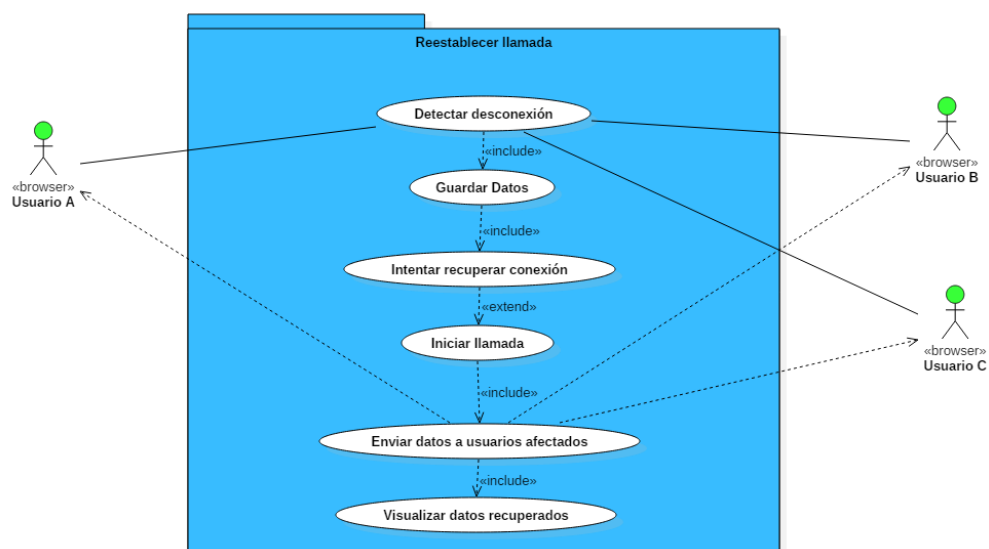
#### Historia de usuario - Grabar audio y video

Historia de Usuario	
Número: 2	Usuario:
Nombre historia: Grabar audio y video	
Prioridad en negocio: Alta (Alta/Media/Baja)	Riesgo en Desarrollo: Alto (Alto, Medio, Bajo)
Puntos Estimados:	Iteración asignada: 1
Programador responsable: Damián Luje – Raúl Llasag	
Descripción: Grabar el contenido de la cámara y el micrófono en tiempo de desconexión.	
Observaciones:	

**Tabla 18**  
**Historia de usuario - Recuperar información**

Historia de Usuario	
Número: 3	Usuario:
Nombre historia: Recuperar información	
Prioridad en negocio: Alta (Alta/Media/Baja)	Riesgo en Desarrollo: Alto (Alto, Medio, Bajo)
Puntos Estimados:	Iteración asignada: 1
Programador responsable: Damián Lujé – Raúl Llasag	
Descripción: Intercambiar el contenido de la cámara y el micrófono grabado en tiempo de desconexión con los interlocutores.	
Observaciones: Brindar la posibilidad de mostrar únicamente la grabación de una desconexión en particular.	

### 3.9.2. Casos de Uso



**Figura 25 Caso de uso - Restablecer llamada**

Tabla 19

**Descripción de caso de uso - Restablecer llamada**

<b>Caso de Uso:</b> Restablecer llamada
<b>Actores:</b> Usuarios
<b>Resumen:</b> si se detecta una desconexión, los usuarios afectados empezarán a almacenar los datos que debían ser transmitidos durante el tiempo de desconexión e intentarán restablecer la comunicación. En cuanto la comunicación se restablezca la llamada se iniciará nuevamente y se enviará la información guardada a los usuarios afectados.
<b>Precondiciones:</b> todos los usuarios deben haber ingresado al sistema, encontrarse en la misma sala de videoconferencia y un interlocutor debe presentar una desconexión de red.
<b>Descripción:</b> <ol style="list-style-type: none"> <li>1. Analizar el estado de la conexión</li> <li>2. Si se detecta la desconexión se iniciará la grabación en la cámara y el micrófono en todos los usuarios afectados por dicha desconexión.</li> <li>3. El usuario desconectado intentará recuperar la conexión</li> <li>4. Cuando la conexión se recupere se iniciará nuevamente la llamada</li> <li>5. Cada usuario envía los datos grabados durante el tiempo de desconexión a los usuarios afectados.</li> <li>6. Cada usuario puede visualizar los datos recuperados.</li> </ol>
<b>Post condiciones:</b>
<b>Observaciones:</b>

**3.9.3. Diagrama de clases**

A la (figura 17) se le ha añadido en el controlador del lado del cliente, las clases de conexión de websockets y el manejador de datos multimedia, descrito por la (figura 26). Éstas proveen de la reconexión del canal de comunicación y de la grabación del contenido de la cámara y el micrófono.

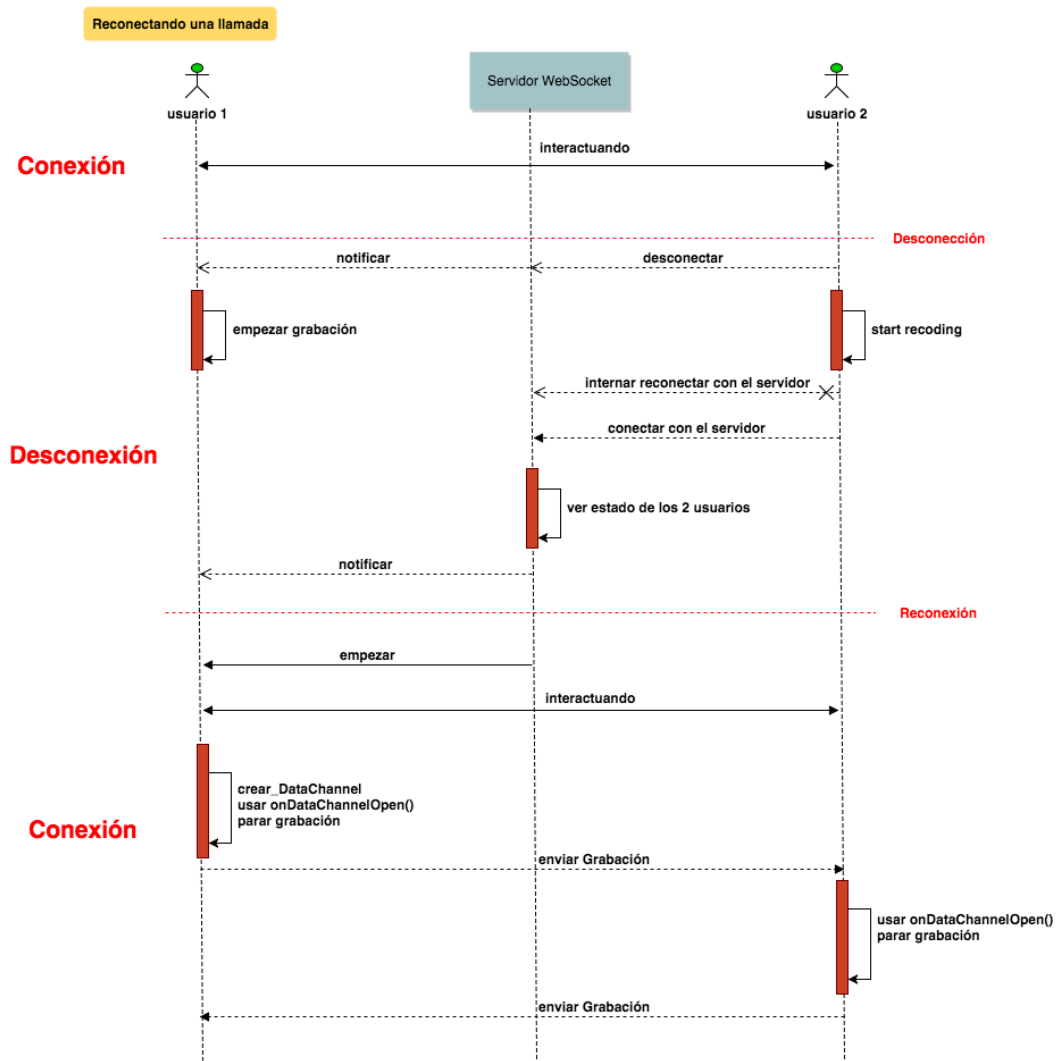


Figura 26 Diagrama de clases de mecanismo de control

**3.9.4. Diagrama de secuencia**

Los dos diagramas de secuencia que se presentan en esta sección son la base de la interacción de los puntos en cuanto alguno de estos presenta una desconexión en la red inalámbrica. El primero, la (figura 27) describe el establecimiento de la reconexión

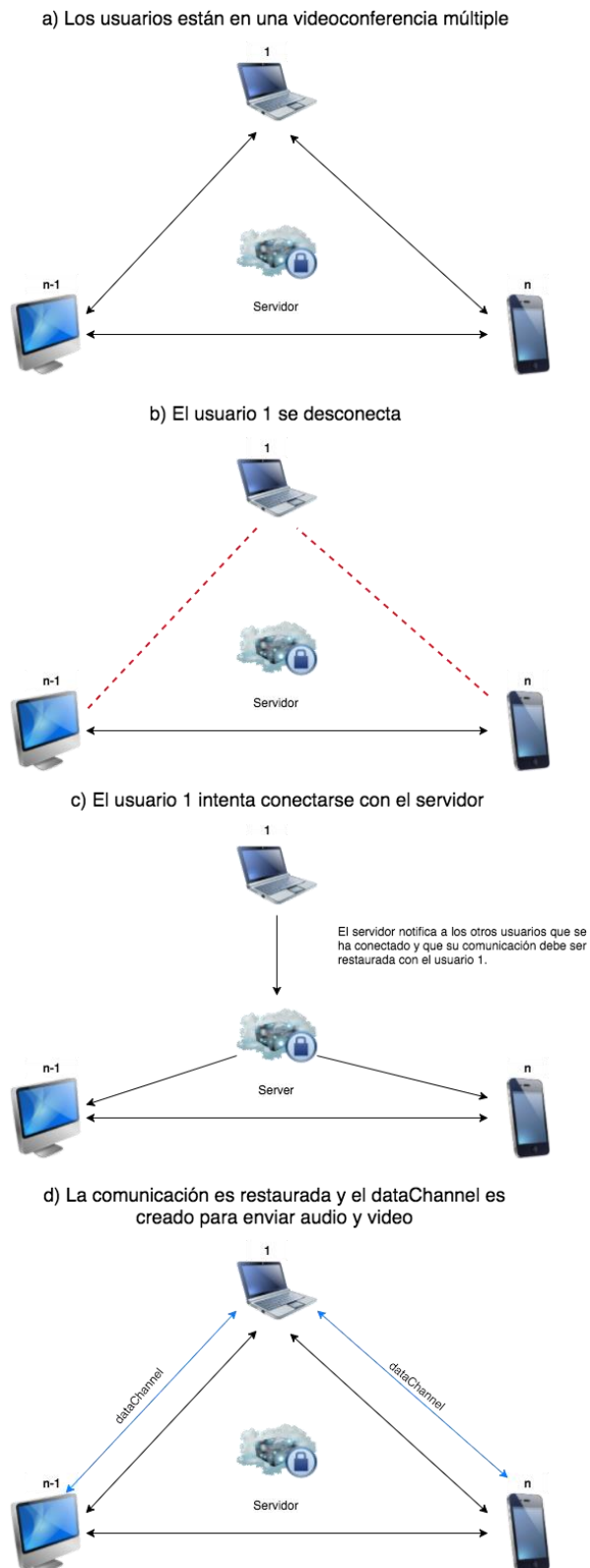
punto a punto de dos usuarios mediante métodos de las APIs RTCPeerConnection y RTCDataChannel de WebRTC y métodos del servidor de websockets. En éste diagrama tiene inicio cuando el usuario 2 es quien se ha desconectado de la red inalámbrica y finaliza en el momento en que vuelven a tener comunicación los puntos.



**Figura 27 Reconexión de una llamada de dos usuarios**

El segundo diagrama descrito por la (figura 28) es una generalización a n usuarios de la desconexión de un punto, este modelo de 4 pasos contiene la interacción directa con el servidor que contiene al controlador de los eventos de la llamada. Si un usuario se desconecta, intenta reconectarse con la ayuda del servidor de websockets, éste comunica al resto de usuarios que se deben conectar a la misma llamada cuando el usuario que se ha desconectado vuelve a tener conexión. Es ahí cuando se crea un canal

de datos (Datachannel) para intercambiar las grabaciones de cada uno de los interlocutores en tiempo de desconexión.



**Figura 28 Reconexión de una videoconferencia**

### 3.9.5. Control de iteraciones

**Tabla 20**

**Control de iteraciones del mecanismo de desconexión**

H U	Tarea	Actividad	Descripción	Desarrollador	Fecha de inicio	Fecha final
1	Reconectar llamada	<ul style="list-style-type: none"> <li>• Detectar si un usuario se desconecta.</li> <li>• Validar si el usuario desconectado se encontraba en una sala de videoconferencia.</li> <li>• Restablecer el canal de comunicación.</li> </ul>	<ul style="list-style-type: none"> <li>• Verificación de una desconexión mediante un indicador.</li> <li>• Visualización de un lapso de pérdida de información.</li> <li>• Visualización de una videoconferencia con los mismos interlocutores.</li> </ul>	Raúl Llasag Damián Luje	2016-09-20	2016-09-30
2	Grabar Audio y video	<ul style="list-style-type: none"> <li>• Grabar contenido de la cámara y el micrófono.</li> </ul>	<ul style="list-style-type: none"> <li>• Desconectar, grabar y almacenar en local-storage del navegador</li> <li>• Finalización de la grabación cuando se haya restablecido la llamada.</li> </ul>	Raúl Llasag Damián Luje	2016-08-30	2016-10-28
3	Recuperar información	<ul style="list-style-type: none"> <li>• Intercambiar grabaciones entre puntos.</li> </ul>	<ul style="list-style-type: none"> <li>• Selección de la información a recuperar.</li> <li>• Visualización la información recuperada.</li> </ul>	Raúl Llasag Damián Luje	2016-10-03	2016-10-28



### 3.9.6. Pruebas Funcionales

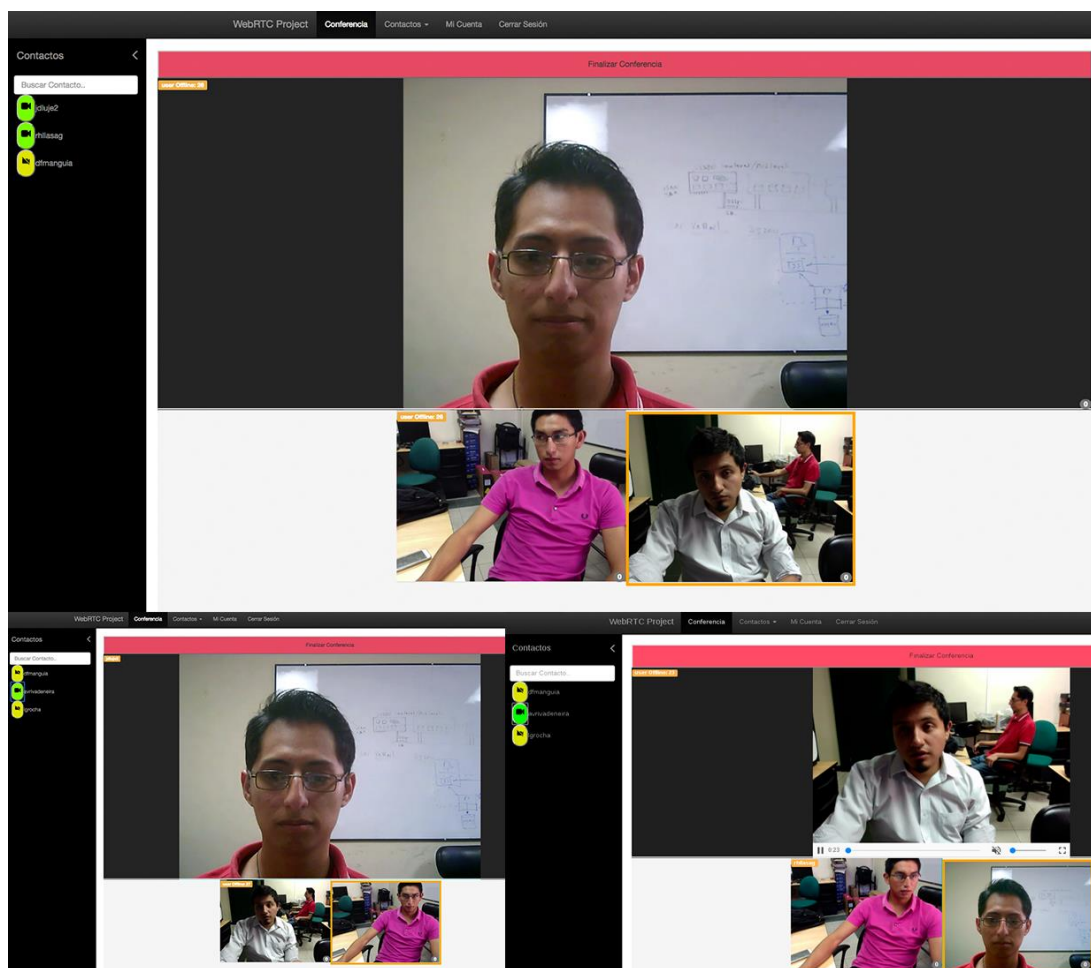


Figura 29 Caso de prueba - Reconectar llamada - Usuario desconectado

Tabla 21

#### Caso de prueba - Restablecer llamada

<b>Campos a diligenciar</b>	Audio y video en videoconferencia
<b>ID Campo de Prueba</b>	PF005
<b>Nombre caso de prueba</b>	Reconectar llamada
<b>Descripción</b>	Se restablece la comunicación de la sala.
<b>Precondiciones</b>	Desconectar y reconectar la red de un usuario durante una llamada.
<b>Relaciones Caso de Uso</b>	Restablecer llamada
<b>Pasos y condiciones de ejecución</b>	<ol style="list-style-type: none"> <li>1. Participar en una llamada.</li> <li>2. Desconectar al equipo de la red.</li> <li>3. Restablecer la conexión.</li> </ol>
<b>Resultado esperado</b>	Una vez que se restablece la conexión se vuelve a visualizar en pantalla la imagen de los usuarios implicados en esa sala de videoconferencia.
<b>Estado caso de prueba</b>	Exitoso
<b>Resultado obtenido</b>	Se ejecuta de acuerdo a lo planeado
<b>Errores asociados</b>	No tener de vuelta la conexión a la red.
<b>Responsable diseño</b>	Raúl Llasag
<b>Responsable ejecución</b>	Damián Lujé

Tabla 22

## Caso de prueba - Grabación de audio y video

<b>Campos a diligenciar</b>	Contenido de micrófono y cámara
<b>ID Campo de Prueba</b>	PF006
<b>Nombre caso de prueba</b>	Grabación de audio y video
<b>Descripción</b>	Se almacena en disco la grabación del contenido de la cámara y el micrófono en momento de desconexión de un miembro de la sala de videoconferencia.
<b>Precondiciones</b>	Desconectar y reconectar la red de un usuario.
<b>Relaciones Caso de Uso</b>	Grabar audio y video
<b>Pasos y condiciones de ejecución</b>	<ol style="list-style-type: none"> <li>1. Participar en una sala de videoconferencia.</li> <li>2. Desconectar al equipo de la red.</li> <li>3. Restablecer la conexión.</li> </ol>
<b>Resultado esperado</b>	
<b>Estado caso de prueba</b>	Exitoso
<b>Resultado obtenido</b>	Se ejecuta de acuerdo a lo planeado
<b>Errores asociados</b>	No tener de vuelta la conexión a la red.
<b>Responsable diseño</b>	Raúl Llasag
<b>Responsable ejecución</b>	Damián Lujé

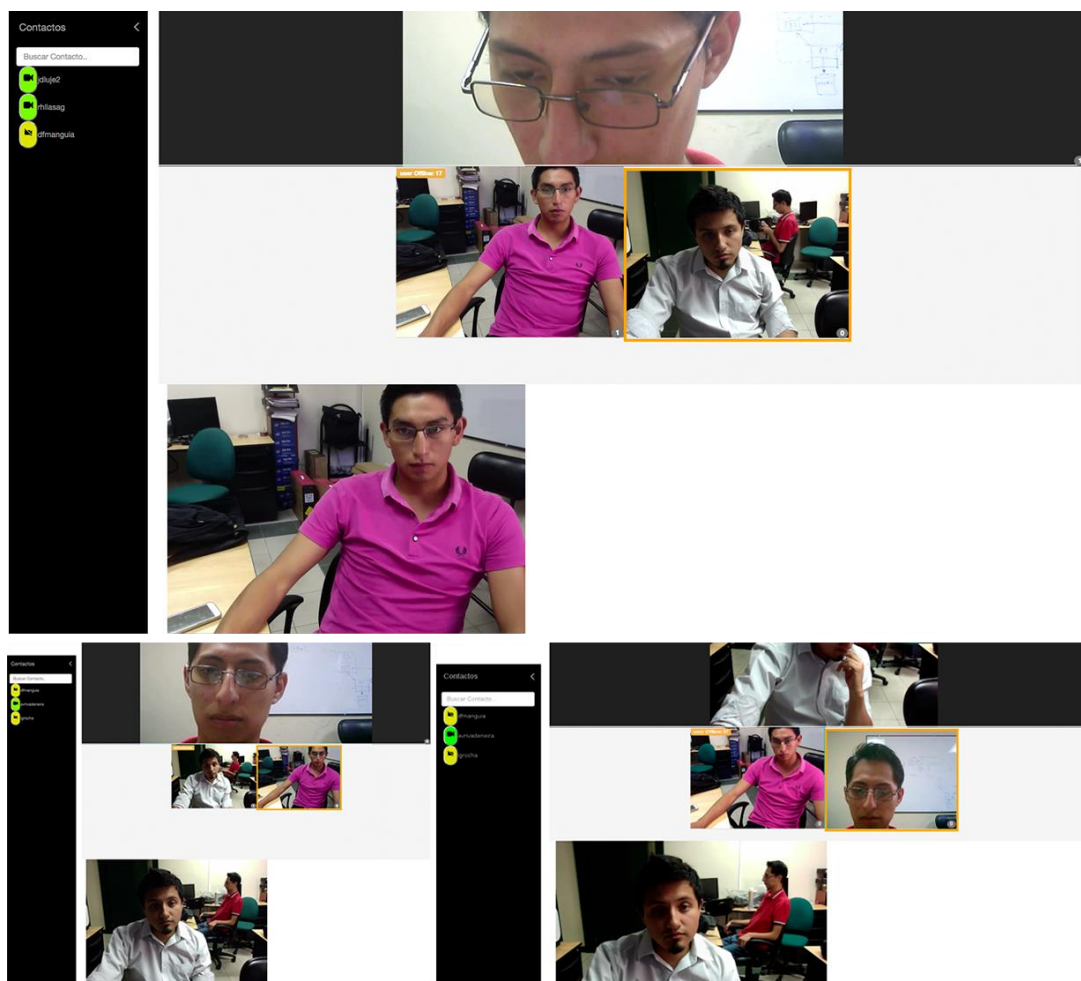


Figura 30: Caso de prueba - Visualizar audio y video

**Tabla 23**  
**Caso de prueba - Visualizar audio y video**

<b>Campos a diligenciar</b>	Audio y video grabado
<b>ID Campo de Prueba</b>	PF007
<b>Nombre caso de prueba</b>	Visualizar audio y video recuperado
<b>Descripción</b>	Se recupera y visualiza la información recuperada del tiempo de desconexión de un usuario de la sala de videoconferencia.
<b>Precondiciones</b>	Desconectar y reconectar la red de un usuario.
<b>Relaciones Caso de Uso</b>	Recuperar información
<b>Pasos y condiciones de ejecución</b>	<ol style="list-style-type: none"> <li>1. Participar en una sala de videoconferencia.</li> <li>2. Desconectar al equipo de la red.</li> <li>3. Restablecer la conexión.</li> <li>4. Escoger la visualización de la información recuperada</li> </ol>
<b>Resultado esperado</b>	El usuario puede visualizar en pantalla la información recuperada.
<b>Estado caso de prueba</b>	Exitoso
<b>Resultado obtenido</b>	Se ejecuta de acuerdo a lo planeado
<b>Errores asociados</b>	No tener de vuelta la conexión a la red.
<b>Responsable diseño</b>	Raúl Llasag
<b>Responsable ejecución</b>	Damián Lujé

### 3.10. Implantación y herramientas base

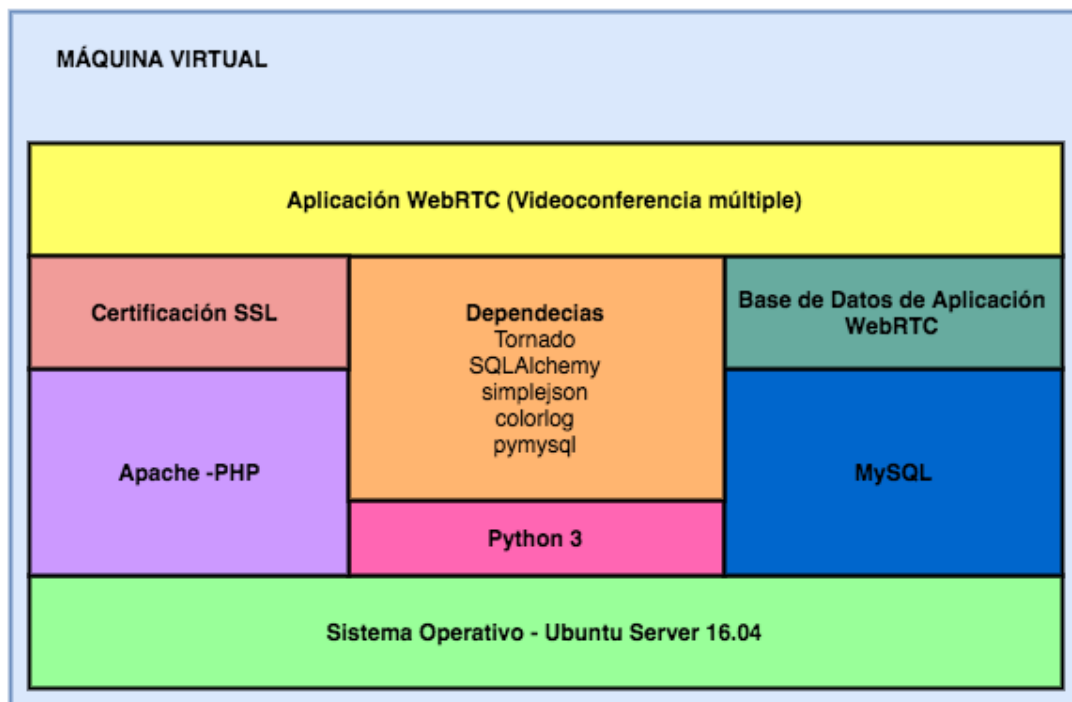
La instalación de la aplicación de videoconferencia múltiple y su mecanismo de control de desconexión están ubicados sobre una máquina virtual alojada en la nube (figura 31). Se le ha instalado un sistema operativo Linux Ubuntu Server 16.04 y el servicio web Apache con PHP, sobre estos un certificado SSL del proyecto Let`s Encrypt.

La necesidad de una gestión de contactos de usuario ha obligado al uso e instalación de la base de datos relacional sobre MySQL, pero la herramienta que en realidad controla el establecimiento de canales de comunicación entre los interlocutores está desarrollada sobre el Lenguaje de Programación Python 3, con las dependencias:

- tornado: gestiona los websockets para obtener información, conectar y reconectar conexiones.
- SQLAlchemy: contiene el acceso a la base de datos.
- simplejson: gestiona objetos JSON provenientes de Javascript.

- colorlog: Brinda una estructura de Registro sobre el servidor de websockets.
- pymysql: gestiona los datos de la base MySQL.

Sobre toda esa plataforma de herramientas se despliega la carpeta de aplicación WebRTC desarrolla y ubicada sobre el directorio del servidor de aplicaciones para la web.



**Figura 31 Diagrama de implantación**

## CAPÍTULO IV

### PRUEBAS DE CALIDAD

#### 4.1. Introducción

El presente capítulo presenta los resultados de las pruebas de calidad de servicio (QoS) y de experiencia de usuario (QoE). Como parámetros de evaluación del servicio de videoconferencia múltiple y su mecanismo de control de desconexión se tiene la evaluación del retardo, jitter y pérdida de información, para audio y video respectivamente. Sus valores se recogen gracias a la implementación del módulo de estadísticas, implementada gracias a la API getStats que interactúa directamente con el registro del navegador webrtc-internals.

El resultado que define la experiencia de usuario se da gracias al uso de la métrica de evaluación directa MOS con participación de usuarios no expertos, un ambiente de pruebas adecuado y el método de calificación de categorías absoluta para audio y video, propuestos por las recomendaciones de la Unión Internacional de Telecomunicaciones.

#### 4.2. Evaluación de calidad en los servicios de streaming

El incremento en la demanda de transmisión de contenidos mediante streaming, ha obligado a sus proveedores a mejorar sus servicios a través de pruebas de calidad, por lo que varios estudios se han acentuado en nuevos conceptos, métricas, parámetros de evaluación, estándares y rangos de valores aceptados por la percepción de usuarios.

La práctica de pruebas de calidad QoS y QoE, hoy en día son obligatorias para ofrecer un servicio competitivo. Sin embargo a pesar que estas pruebas son complementarias para la obtención de un valor de aceptación concreto por los usuarios, éstas son muy costosas y en muchos de los casos se tienen que realizar manualmente.

Integrar la medición de QoS y QoE es un reto, los servicios de streaming, las arquitecturas sobre las que se soportan, las topologías de comunicación, las ilimitadas condiciones de uso de las herramientas, por esta razón existen herramienta software de medición para ambientes específicos, como por ejemplo: Perceptual Evaluation of

Video Quality (PEVQ™) que se basa en varios estándares de Telecommunication Standardization Sector (ITU-T) para emitir un valor de MOS del servicio de streaming de video en las plataformas de Linux y Windows (Opticom, 2016), o también la herramienta de código abierto BOX, que emula distintos escenarios de red existente y emite un criterio de percepción del servicio de streaming de video (Carrera, 2015).

Debido a la inexistencia de una herramienta que integre y automatice las pruebas experiencia de usuario y de calidad del servicio de videoconferencia múltiple sobre un sistema distribuido P2P híbrido, se plantean casos de evaluación especiales, con el fin de aceptar o rechazar la hipótesis planteada en ésta investigación, así como también valorar la competitividad de la plataforma con el uso de la métrica MOS.

### 4.3. Definición de casos de prueba de calidad de servicio

Para valorar la calidad de servicio QoS, se definen 3 casos de prueba, como lo describe la (tabla 25). Gracias al uso del módulo de vista de estadísticas, se recogen valores de variación del retardo, jitter y bytes transmitidos que se presentan en una duración de 3 minutos, con desconexiones de aproximadamente 15 segundos, períodos en los cuales es óptima realizar las pruebas.

**Tabla 24**  
**Casos de evaluación**

<i>Caso de evaluación</i>	<i>Número de interlocutores</i>	<i>Número de interconexiones</i>	<i>Número de desconexiones</i>	<i>Duración</i>
<b>CE001</b>	2	1	1	3 min
<b>CE002</b>	2	2	3	3 min
<b>CE003</b>	3	3	3	3 min

Los casos de prueba se dividen principalmente por la cantidad de interlocutores. Por cada interlocutor  $n$ , existe un número de interconexiones  $C(n, 2)$  entre puntos, y gracias a la definición de un modelo matemático generalizado de estudio de eventos, se evalúa el desempeño en cada interconexión.

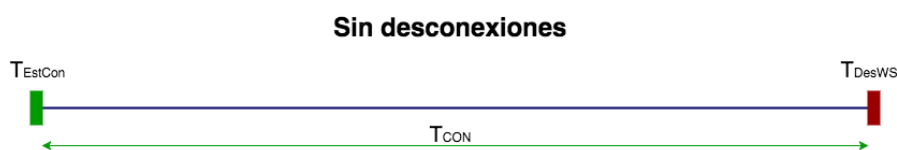
#### 4.4. Modelo Matemático con variables de tiempo

##### 4.4.1. Importancia

La definición de un modelo matemático para la evaluación del desempeño del servicio de videoconferencia múltiple surge a partir de la necesidad de medir la información perdida en los tiempos de desconexión de la Red de un usuario y por dos impedimentos tecnológicos. El primer impedimento es la inexistencia del análisis de información enviada por la API Datachannel y la segunda es la diferencia de dimensiones de paquetes de los codecs VP8 y VP9 en la transmisión de video en tiempo de conexión y desconexión respectivamente. Por estas razones se hace inútil el uso de una herramienta que cense el tráfico de paquetes de los protocolos de comunicación como Wireshark o Tcpcdump y se propone el análisis de información perdida mediante el análisis de periodos de tiempo.

##### 4.4.2. Definición

El ciclo de vida normal de una video llamada se da origen en tiempo  $T_{EstCon} = 0$  con un el establecimiento de la conexión de RTCPeerConneccion y culmina con un tiempo final al identificar la desconexión del websocket con el servidor  $T_{DesWS}$  como lo describe la (figura 32). En el este caso, el valor del período de tiempo total de la llamada es igual al período de conexión y a la variable de la desconexión del websocket con el servidor  $T_{TOTAL} = T_{DesWS} = T_{CON}$ .

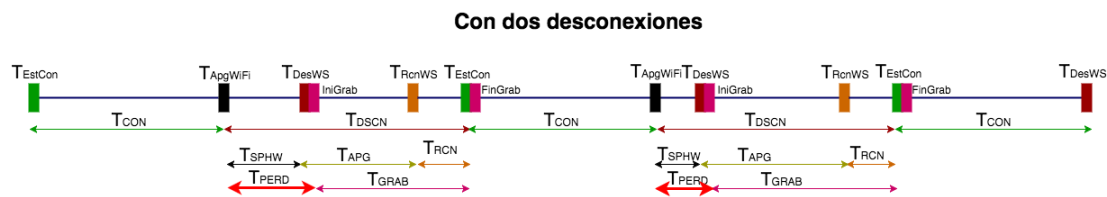


**Figura 32 Ciclo de vida sin desconexiones**

Si, al menos un usuario se va de la Red, se produce un período de tiempo de desconexión  $T_{DSCN}$ , el cual se puede calcular gracias a las variables  $T_{AppWiFi}$  y  $T_{EstCon}$ . No se asegura la recuperación de la información en su totalidad en el lapso  $T_{DSCN}$ , por este motivo el cálculo del error contiene las variables  $T_{DesWS}$  y  $T_{RcnWS}$ , y los períodos  $T_{APG}$ ,  $T_{RCN}$  y  $T_{GRAB}$  que se observan en la (figura 33) y se describen en la (tabla 26).

La obtención de estos datos es automatizada en el módulo de estadísticas de la aplicación, excepto por  $T_{AppWiFi}$  que se toma manualmente del Registro de conexión de

red inalámbrica del sistema operativo. Y gracias a este análisis se llega a concluir que entre la desconexión manual de la Red ( $T_{AppWiFi}$ ) y el instante que se da cuenta el servidor de esto, existe un período  $T_{SPHW}$ , el cual es diferente al total de segundos perdidos  $T_{PERD}$ , porque en el primer caso interviene en su cálculo  $T_{APG}$  y  $T_{RCN}$ , y en el segundo  $T_{GRAB}$  que no tiene ninguna relación con los dos anteriores, pues la grabación inicia después de la notificación de desconexión, incrementando el error total. Este análisis también se puede visualizar a más detalle en la (tabla 25).



**Figura 33 Ciclo de vida con dos desconexiones**



**Tabla 25**  
**Definición de variables de tiempo**

<b>Nombre</b>	<b>Nomenclatura</b>	<b>Tipo</b>	<b>Cálculo</b>	<b>Descripción</b>
Establecimiento de conexión.	$T_{EstCon}$	Evento	-	-Establece la comunicación P2P con RTCPeerConnection.
Desconexión manual de Red	$T_{AppWifi}$	Evento	-	-Se da cuando se desconecta manualmente la conexión Wifi.
Desconexión del websocket	$T_{DesWS}$	Evento	-	-Establece la conexión con el servidor.
Reconexión del websocket	$T_{RcnWS}$	Evento	-	-Desconecta la comunicación con el servidor.
Ecuación 3 Conexión	$T_{CON}$	Período de tiempo	$T_{AppWiFi} - T_{EstCon}$	-El período desde que se establece la comunicación P2P hasta la desconexión manual.
Ecuación 4 Desconexión	$T_{DSCN}$	Período de tiempo	$T_{EstCon} - T_{AppWifi}$	-Desde la desconexión del websocket hasta el establecimiento de la comunicación P2P.
Ecuación 5 Tiempo perdido por hardware	$T_{SPHW}$	Período de tiempo	$T_{DesWS} - T_{AppWifi}$	-Desde la desconexión manual hasta que se desconecta el websocket.
Ecuación 6 Tiempo fuera de websocket	$T_{APG}$	Período de tiempo	$T_{RcnWS} - T_{DesWS}$	-Lapso en que el punto no se comunica con el servidor mediante el websocket.
Ecuación 7 Reconexión	$T_{RCN}$	Período de tiempo	$T_{EstCon} - T_{RcnWS}$	-Desde que se reconecta el websocket hasta que se establece la comunicación P2P.
Ecuación 8 Grabación	$T_{GRAB}$	Período de tiempo	$FinGrab - IniGrab$	-El tiempo de grabación del video en tiempo de desconexión.
Ecuación 9 Total de segundos perdidos	$T_{PERD}$	Período de tiempo	$T_{DSCN} - T_{GRAB}$	-La Diferencia entre el período de desconexión y el tiempo de grabación.
Total	$T_{TOTAL}$	Período de tiempo	(ecuación 9)	-La duración total de la videoconferencia.

Generalizado el modelo a n desconexiones tenemos:

$$T_{TOTAL} = \sum_{1}^{n+1} T_{CON\ n} + \sum_{1}^{n} T_{DSCN\ n}$$

**Ecuación 10 Primera fórmula generalizada del tiempo total**

Sumando los períodos de tiempo

$$\begin{aligned}
 T_{SPHW\ n} &= T_{DesWS\ n} - T_{ApgWiFi\ n} \\
 T_{OFF\ n} &= T_{RcnWS\ n} - T_{DesWS\ n} \\
 + \quad T_{RCN\ n} &= T_{EstCon\ n+1} - T_{RcnWS\ n}
 \end{aligned}$$


---

$$T_{SPHW\ n} + T_{APG\ n} + T_{RCN\ n} = T_{EstCon\ n+1} - T_{ApgWiFi\ n}$$

Sustituyendo en la (ecuación 3) se concluye que:

$$T_{SPHW\ n} + T_{APG\ n} + T_{RCN\ n} = T_{DSCN\ n}$$

Y reemplazando en la (ecuación 9) se obtiene:

$$T_{TOTAL} = \sum_1^{n+1} T_{CON\ n} + \sum_1^n T_{SPHW\ n} + T_{APG\ n} + T_{RCN\ n}$$

### **Ecuación 11 Segunda fórmula generalizada del tiempo total**

Finalmente para obtener la tercera fórmula generalizada se despeja  $T_{DSCN\ n}$  de:

$$T_{PERD\ n} = T_{DSCN\ n} - T_{GRAB\ n}$$

Dando como resultado

$$T_{PERD\ n} + T_{GRAB\ n} = T_{DSCN\ n}$$

Y reemplazando en (ecuación 9) se obtiene:

$$T_{TOTAL} = \sum_1^{n+1} T_{CON\ n} + \sum_1^n T_{PERD\ n} + T_{GRAB\ n}$$

### **Ecuación 12 Tercera fórmula generalizada del tiempo total**

## 4.5. Estándares de Evaluación y valores aceptables

### 4.5.1. Audio

La recomendación de UIT-T G.114 menciona que si los “retardos se mantuvieran por debajo de 150 ms, entonces no se afectaría de modo significativo a la mayoría de las aplicaciones” (UIT, 2003). También establece un límite superior de 400 ms para la planificación de redes, pero menciona que también se pueden dar retardos a nivel de aplicación boca-oído que no son fáciles de aplicar en ésta recomendación.

Para el caso del jitter y uso de ancho de banda, los colaboradores de Cisco (Lewis & Pickavance, 2006), menciona valores aceptables que se detallan en la (tabla 27).

**Tabla 26**  
**Parámetros de calidad de audio**

Parámetro de calidad de servicio	Rango aceptable
Retardo	Óptimo entre 150ms y 400ms
Jitter	Menor a 30ms
Ancho de banda	17000 a 106000 bytes por segundo por prioridad de llamada
Paquetes perdidos	Menor a 1%

### 4.5.2. Video

Para el aprovisionamiento del tráfico en videoconferencia, los colaboradores de Cisco (Lewis & Pickavance, 2006) recomiendan las siguientes pautas:

- Una vía de retardo no debe ser mayor de 400 ms.
- Jitter no debe ser mayor de 30 ms.

**Tabla 27**  
**Parámetros de calidad de video**

Parámetro de calidad de servicio	Rango aceptable
Retardo	Óptimo entre 150ms y 400ms
Jitter	Menor a 30ms
Paquetes perdidos	Menor a 1%

#### 4.6. Entorno

Las pruebas se han realizado sobre los equipos que se describen en la (tabla 28), donde el equipo iMac es el cual desconecta manualmente su red inalámbrica.

**Tabla 28**  
**Equipos usados en las pruebas de calidad**

Equipo	Procesador	Memoria RAM	Sistema Operativo	Navegador	Definición de la cámara web
<b>iMac</b>	3.1 GHz Intel core i5	4GB	OS X 10.2.2	Google Chrome 54.0.2	4096 x 2304
<b>iMac 2</b>	3.1 GHz Intel core i5	4GB	OS X 10.2.2	Google Chrome 54.0.2	4096 x 2304
<b>NUC</b>	1.3 GHz Intel Core i5	4GB	Fedora 24 64bits	Google Chrome 54.0.2	640 x 480

#### 4.7. Casos de Evaluación de videoconferencia y análisis de resultados de pruebas de calidad de servicio

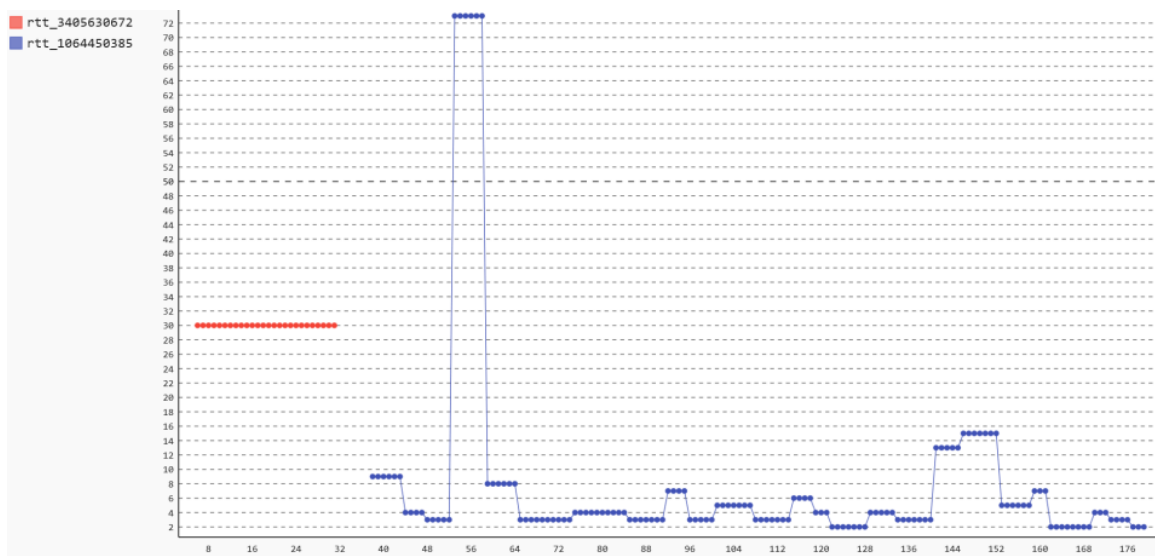
Los casos de evaluación CE001 y CE002 tienen dos interlocutores, la duración de la videoconferencia en teoría es de 3 minutos. CE001 posee una desconexión al segundo diez del primer minuto y una reconexión al segundo veinte y cinco. Mientras que CE002 posee desconexiones a los diez segundos del minuto uno, dos y tres, y reconexiones al segundo veinte y cinco del minuto respectivo, sin embargo estos valores no son posibles de alcanzar manualmente por un retraso ojo-mano.

Se evalúan los canales de comunicación entre pares que interactúan con el punto que se desconecta. Por cada canal de comunicación se evalúan los datos de los dos interlocutores, la información del usuario que desconecta su red se calculan con la

segunda y tercera fórmula generalizada, mientras que el punto que detecta la desconexión analiza sus datos con la primera y tercera.

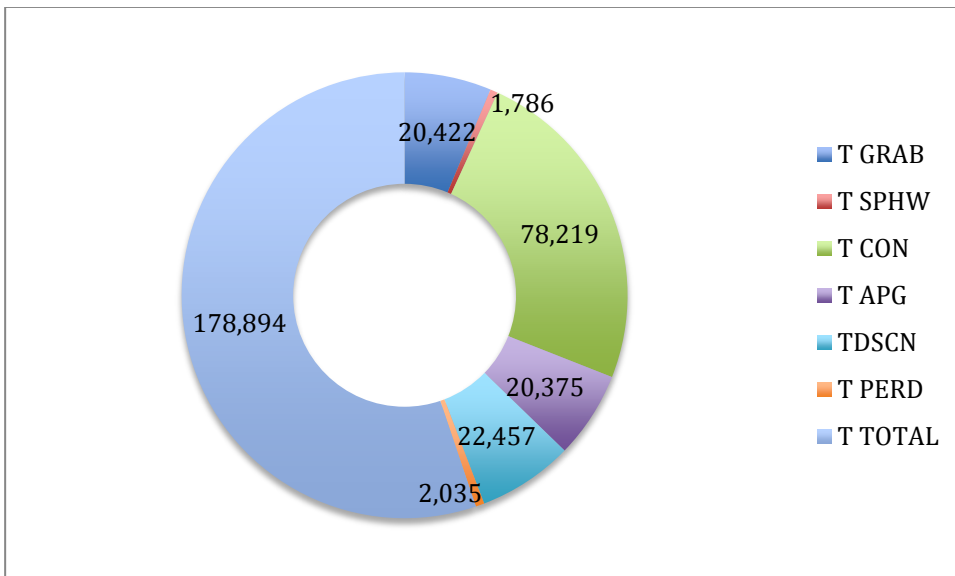
#### 4.7.1.CE001

Éste caso de evaluación con una desconexión tiene como referencia la (figura 34), en la cual se puede observar el establecimiento de la conexión entre pares mediante un indicador de estatus de conexión RTT de webRTC. Cada color significa una conexión y el espacio vacío una desconexión en el tiempo.

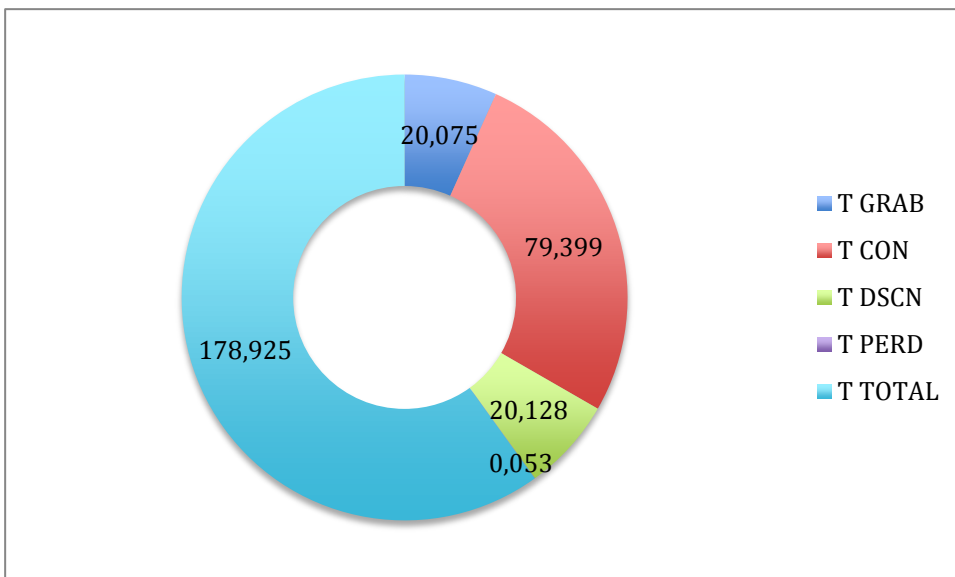


**Figura 34 RTT en CE001**

Con el módulo de estadísticas y una tabulación de 5 repeticiones del caso CE001, se ha llegado a obtener períodos de tiempo que convergen como lo describe la (figura 35) para el equipo iMac que desconecta manualmente la red inalámbrica y la (figura 36) para el NUC que detecta la desconexión.



**Figura 35 iMac desconectado en CE001**



**Figura 36 NUC que detecta desconexión en CE001**

Las estadísticas del retraso, jitter y bytes transmitidos de iMac se describen en la en la (tabla 29) y en la (tabla 30) del NUC. La media del retraso oscila entre 44ms y 126 ms, estando dentro del rango óptimo que propone la UTI-T G.114 para el audio, así como también Cisco para el video. El jitter no superan los 30 ms en los dos casos y el rango aceptable de bytes por segundo no es mayor a 160000 para audio.

**Tabla 29**  
**Audio y Video de iMac en el caso CE001**

Recibido						Enviado		
Video			Audio			Video	Audio	
Retraso	BytesPorSec	Jitter Buffer	Retraso	Jitter Recibido	BytesPorSec	BytesPerSec	Jitter Recibido	BytesPorSec
Media	Media	Media	Media	Media	Media	Media	Media	Media
54,1	154450,6	41,9	73,9	5,9	4341,6	213993,8	4,4	5237,6
77,6	186049,0	62,4	116,4	6,7	4272,8	199890,1	5,3	5239,7
44,0	179047,0	23,4	65,1	4,1	4359,1	213488,3	4,1	5172,6
50,3	189398,7	37,8	72,7	4,8	4368,8	212023,4	4,6	5167,1
50,8	177525,3	31,6	72,3	3,4	4465,4	213544,0	3,4	5181,2

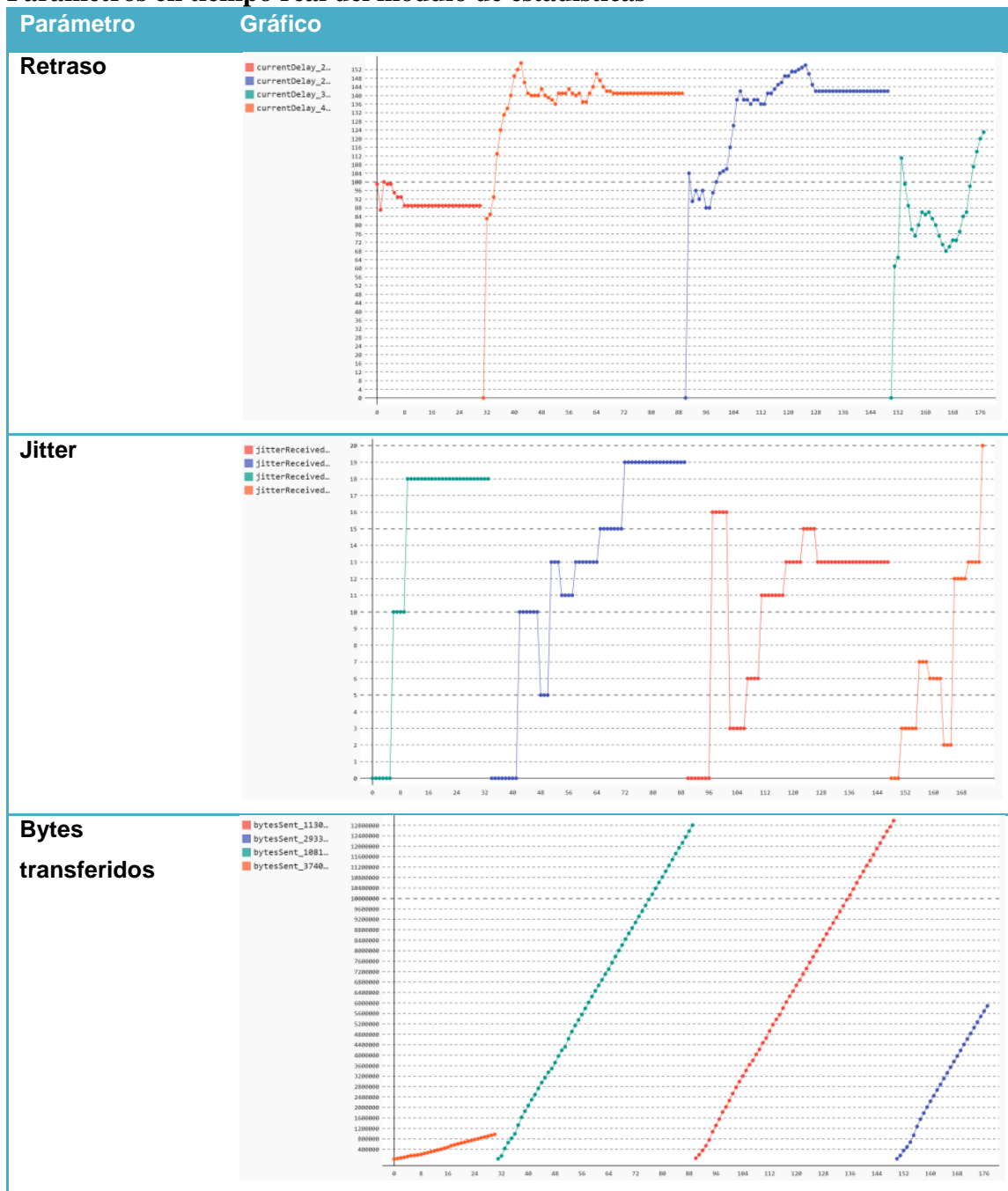
**Tabla 30**  
**Audio y Video en NUC en el caso CE001**

Recibido						Enviado		
Audio			Video			Audio	Video	
Retraso	Jitter Recibido	Retraso	Jitter Recibido	Retraso	Jitter Recibido	Retraso	Jitter Recibido	Retraso
Media	Media	Media	Media	Media	Media	Media	Media	Media
106,9	187613,4	70,6	122,4	4,3	4584,5	161264,2	5,7	4917,5
109,2	172162,8	73,6	126,0	5,6	4536,2	212231,8	6,8	4858,7
87,2	185696,0	66,0	93,4	4,0	4502,1	186305,7	3,9	4957,6
88,8	185720,7	62,7	99,2	4,7	4539,1	215834,9	5,5	4942,5
79,6	188577,4	56,2	99,9	3,0	4585,4	202405,0	4,0	5001,8

#### 4.7.2.CE002

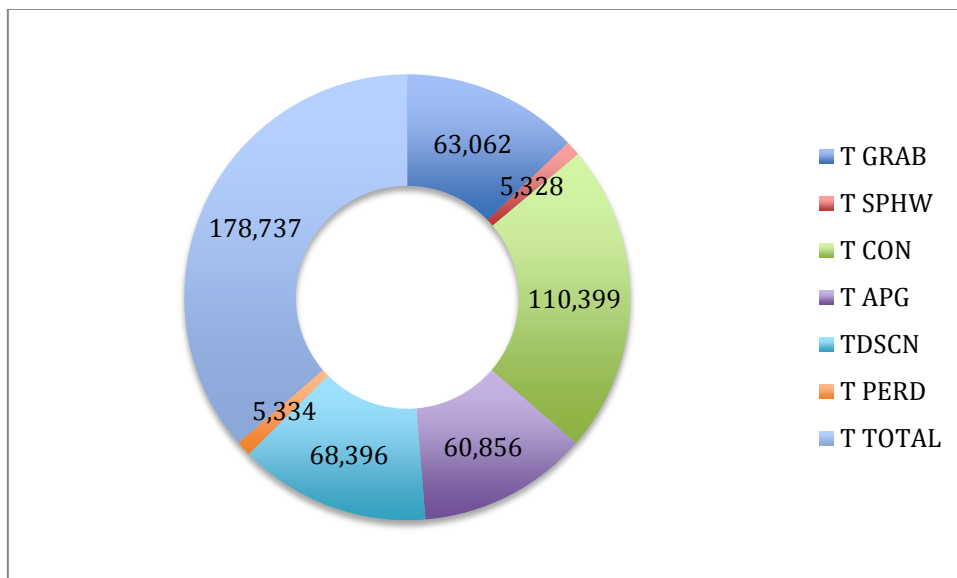
El caso CE002 tiene tres desconexiones, al igual que el caso CE003. Ambos tienen como referencia a un ciclo de vida similar al que indica la (tabla 31), donde existen 3 desconexiones y 4 conexiones. El módulo de estadísticas tiene la capacidad de indicar y almacenar en formato JSON la variación en el tiempo de retardo, jitter, bytes transferidos, estatus de conexión de audio y video.

**Tabla 31**  
**Parámetros en tiempo real del módulo de estadísticas**

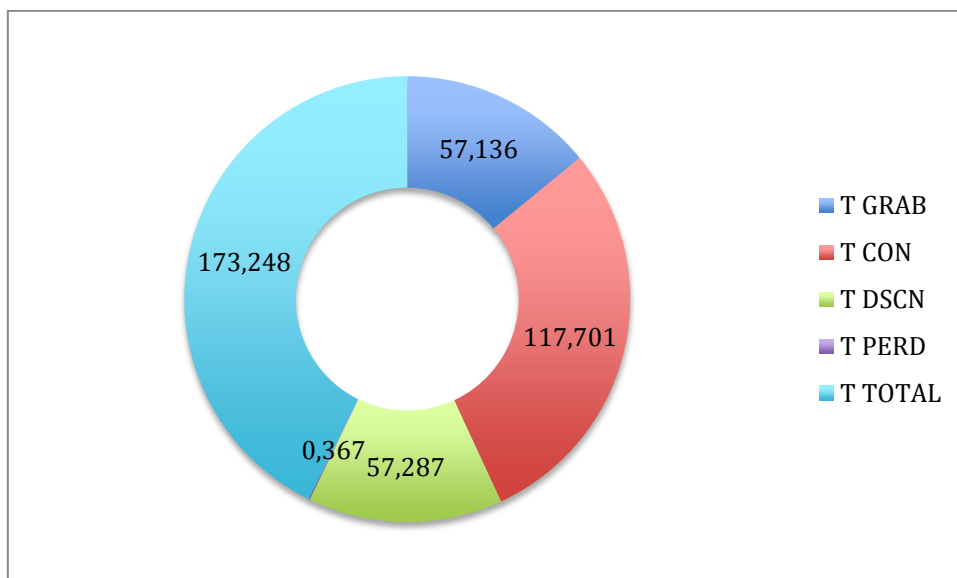


Con la tabulación de 5 repeticiones del caso CE002, se ha llegado a obtener períodos de tiempo que convergen como lo describe la (figura 37) para el equipo iMac que desconecta manualmente la red inalámbrica y la (figura 38) para el NUC que detecta la desconexión.





**Figura 37 iMac desconectado en CE002**



**Figura 38 NUC que detecta desconexión en CE002**

Las estadísticas del retraso, jitter y bytes transmitidos de iMac se describen en la en la (tabla 32) y en la (tabla 33) del NUC. La media del retraso oscila entre 61ms a 251ms para video y de 131ms a 216ms en audio, siendo un rango aceptable para el audio. El jitter no superan los 30 ms en los dos casos y el rango aceptable de bytes por segundo no es mayor a 160000 para audio.

**Tabla 32**  
**Audio y video de iMac en el caso CE002**

Recibido						Enviado		
Audio			Video			Audio	Video	
Retraso	Jitter Recibido	Retraso	Jitter Recibido	Retraso	Jitter Recibido	Retraso	Jitter Recibido	Retraso
Media	Media	Media	Media	Media	Media	Media	Media	Media
61,7	119518,4	48,2	90,9	8,1	2903,6	204776,7	6,6	5030,3
76,2	126489,1	63,9	102,6	9,2	2914,5	206500,6	7,9	5097,9
86,6	119234,3	74,0	104,8	10,9	2919,8	209403,1	10,0	5097,2
113,7	78452,1	123,5	131,4	11,1	2693,1	153053,0	8,4	5051,5
87,2	92878,0	63,3	114,2	10,1	2895,6	209467,5	7,3	5029,3

**Tabla 33**  
**Audio y Video de NUC en el caso CE0002**

Recibido						Enviado		
Video			Audio			Video	Audio	
Jitter Recibido	Retraso	Jitter Recibido	Retraso	Jitter Recibido	Retraso	Jitter Recibido	Retraso	Jitter Recibido
Media	Media	Media	Media	Media	Media	Media	Media	Media
98,3	127106,0	66,4	131,5	7,4	3148,2	180452,3	7,4	4622,4
117,8	128591,5	86,3	120,3	7,7	3177,6	183486,0	11,2	4694,0
119,6	131585,0	81,8	141,1	8,9	3222,2	194165,2	10,9	4693,5
251,3	85024,4	236,2	216,7	10,1	2900,3	177008,7	10,4	4495,1
125,6	131696,4	90,6	131,2	9,1	3177,6	163327,9	8,5	4683,6

#### 4.8. Casos de Evaluación de videoconferencia múltiple y análisis de resultados de calidad de servicio

En la evaluación de la videoconferencia múltiple intervienen 3 canales de comunicación:

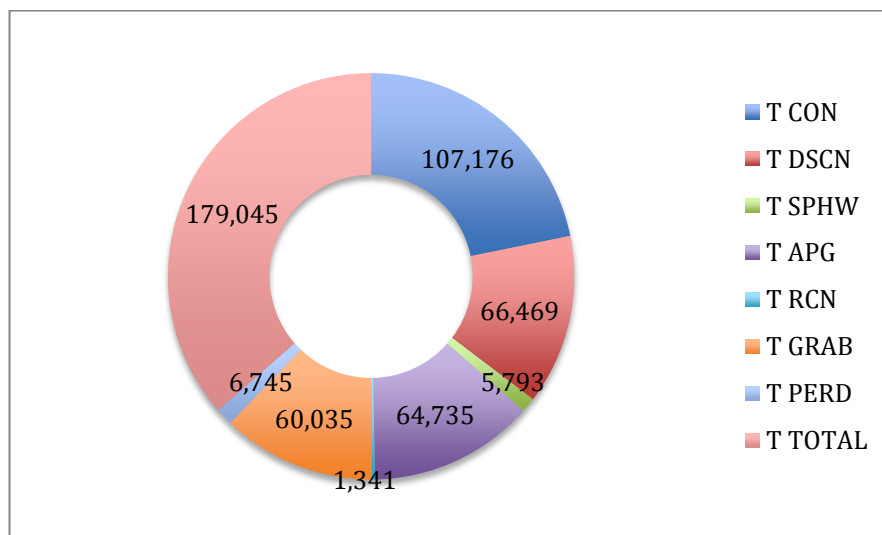
- iMac - NUC
- iMac - iMac 2
- NUC - iMac 2

De los cuales solamente los dos primeros se utilizan para el análisis de datos, puesto que el caso de evaluación tiene por condición la comunicación continua entre iMac 2 – NUC. El inicio de la prueba se da al incluir a iMac en la comunicación de los otros equipos y finaliza cuando iMac deja de participar en la videoconferencia.

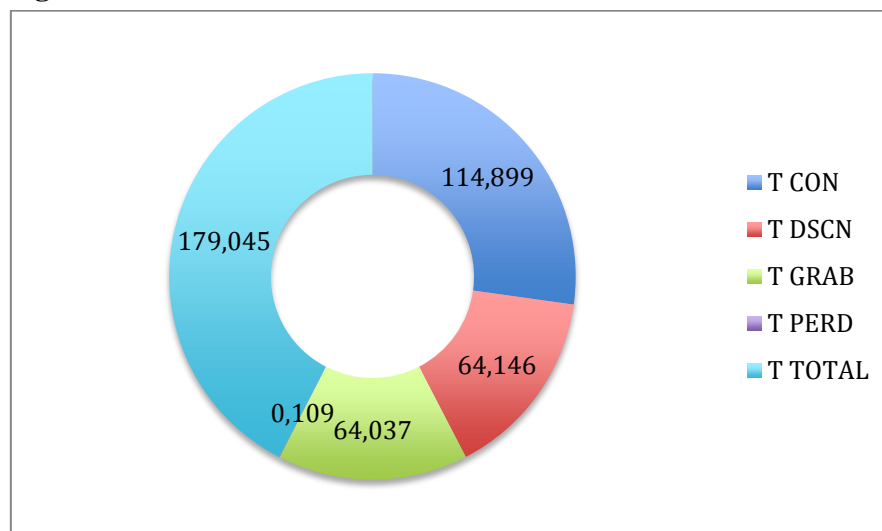
#### 4.8.1.CE003

Con la tabulación de 5 repeticiones del caso CE003, se ha llegado a obtener períodos de tiempo que convergen como lo describe:

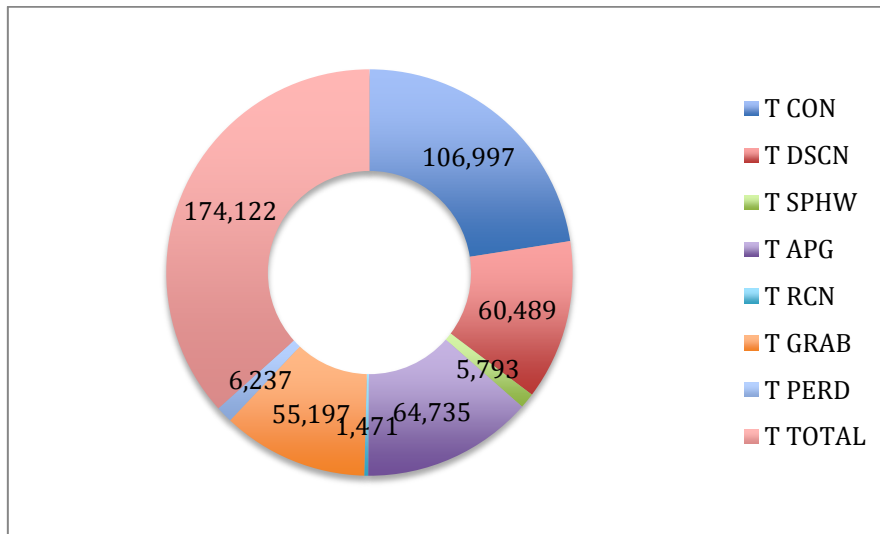
- La (figura 39) para el equipo iMac que desconecta manualmente la red inalámbrica con el NUC.
- La (figura 40) para el equipo NUC que detecta la desconexión con iMac.
- La (figura 41) para el equipo iMac que desconecta manualmente la red inalámbrica con iMac 2.
- La (figura 42) para el equipo iMac 2 que detecta la desconexión con iMac.



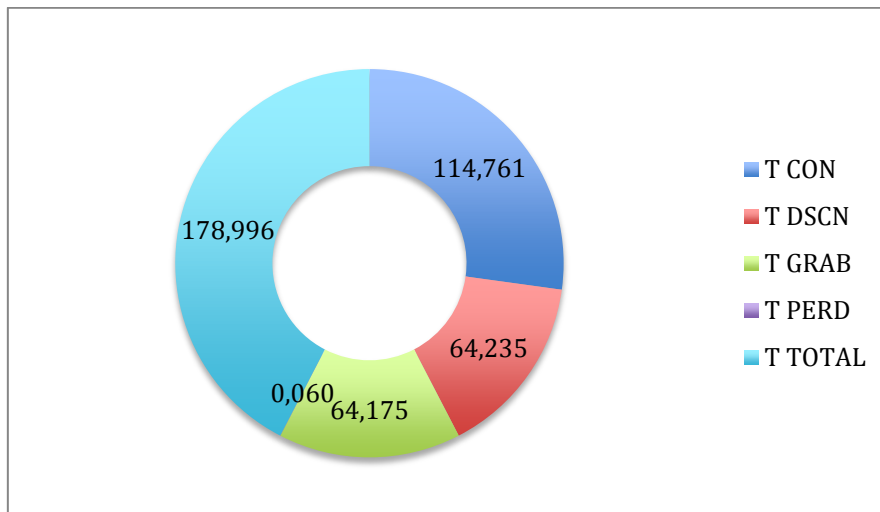
**Figura 39 iMac desconectado en CE003 con NUC**



**Figura 40 NUC que detecta desconexión en CE003 de iMac**



**Figura 41 iMac desconectado en CE003 con iMac 2**



**Figura 42 iMac 2 que detecta desconexión en CE003 de iMac**

Las estadísticas del retraso, jitter y bytes transmitidos de iMac, iMac 2 y NUC se describen en la en la (tabla 34, 35, 36 y 37). La media del retraso oscila entre 113 a 176 ms para video y de 92 a 179 en audio, siendo un rango aceptable para el audio. El jitter no superan los 30 ms en los dos casos y el rango aceptable de bytes por segundo no es mayor a 160000 para audio.

**Tabla 34****Audio y Video de iMac que desconecta de NUC en el caso CE003**

Recibido						Enviado		
Video			Audio			Video	Audio	
Jitter Recibido	Retraso	Jitter Recibido	Retraso	Jitter Recibido	Retraso	BytesPerSec	Jitter Received	BytesPerSec
Media	Media	Media	Media	Media	Media	Media	Media	Media
131,5	108077,6	69,1	169,3	12,1	2782,3	193906,9	10,1	5079,5
113,7	104643,0	101,3	131,1	13,0	2891,2	153901,2	10,8	4812,8
115,2	77444,9	91,3	134,6	11,6	2826,8	160027,0	10,4	5035,7
114,9	78358,5	115,2	138,3	11,2	2765,3	177514,2	10,6	5076,9
114,1	97094,5	99,7	163,1	14,7	2625,9	199678,0	12,2	5141,9

**Tabla 35****Audio y Video de NUC que detecta a iMac en el caso CE003**

Recibido						Enviado		
Video			Audio			Video	Audio	
Jitter Recibido	Retraso	Jitter Recibido	Retraso	Jitter Recibido	Retraso	Jitter Recibido	Retraso	Jitter Recibido
Media	Media	Media	Media	Media	Media	Media	Media	Media
119,4	108335,0	80,6	139,2	10,7	2964,5	193490,1	13,7	4672,4
130,1	92503,3	95,7	131,6	10,5	3024,7	174467,0	14,5	4637,4
126,7	83943,0	97,6	135,8	10,0	3110,9	148045,4	12,9	4726,7
109,0	95645,4	78,8	137,7	11,1	3105,9	122935,8	13,4	4646,8
123,5	109468,6	90,5	147,6	12,9	2870,4	168952,3	15,0	4700,5

**Tabla 36****Audio y Video de iMac que desconecta de iMac 2 en el caso CE003**

Recibido						Enviado		
Video			Audio			Video	Audio	
Jitter Recibido	Retraso	Jitter Recibido	Retraso	Jitter Recibido	Retraso	BytesPerSec	Jitter Received	BytesPerSec
Media	Media	Media	Media	Media	Media	Media	Media	Media
92,3	80499,0	78,3	114,3	11,6	2946,2	169661,1	11,8	4971,0
117,3	88228,7	105,9	117,8	11,5	3025,4	189003,0	11,9	4788,5
109,9	97241,9	98,2	113,1	9,9	3106,7	199614,3	12,3	4993,6
89,4	83021,1	77,2	119,0	10,3	3144,4	194419,1	9,9	5153,7
119,0	51569,1	107,4	122,4	14,0	2927,7	230158,8	13,1	5129,0

**Tabla 37**  
**Audio y Video de iMac 2 que detecta a iMac en el caso CE003**

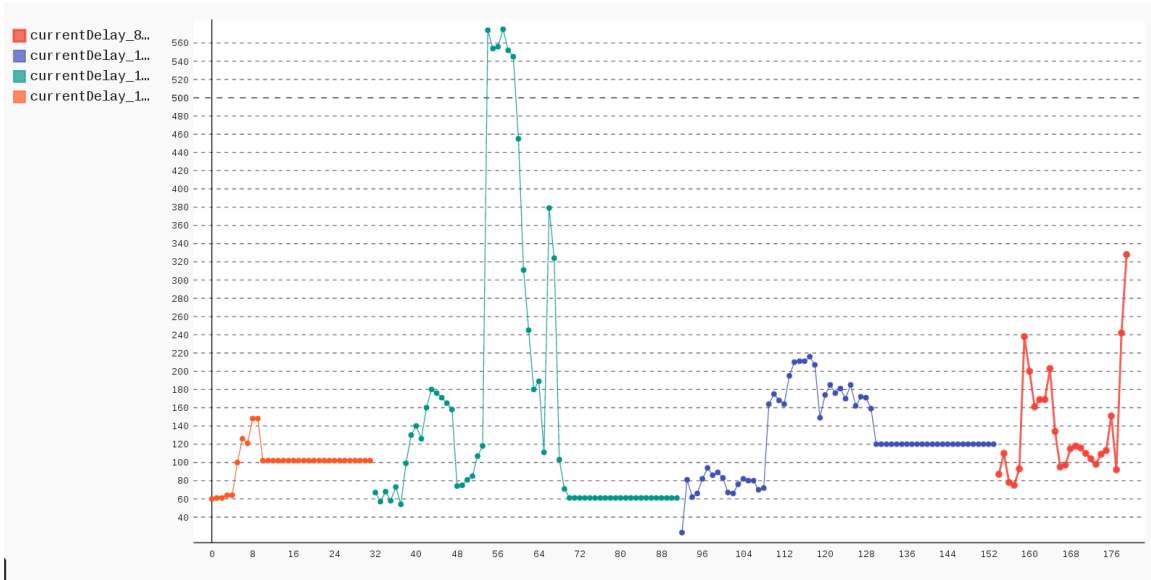
Recibido						Enviado		
Video			Audio			Video	Audio	
Jitter Recibido	Retraso	Jitter Recibido	Retraso	Jitter Recibido	Retraso	BytesPerSec	Jitter Received	BytesPerSec
Media	Media	Media	Media	Media	Media	Media	Media	Media
154,8	97701,2	134,4	164,6	11,3	2947,9	163408,5	13,1	4996,5
123,1	119679,7	110,4	126,7	9,7	3009,3	166550,7	12,6	4809,2
179,5	116630,5	168,3	136,8	9,9	3073,8	178296,7	13,4	5033,2
170,5	117841,8	126,1	176,9	9,9	3112,6	165216,0	10,3	5129,3
151,7	114987,0	110,6	151,0	13,1	2915,6	122118,2	11,8	5141,5

#### 4.9. Análisis de variación de parámetros QoS

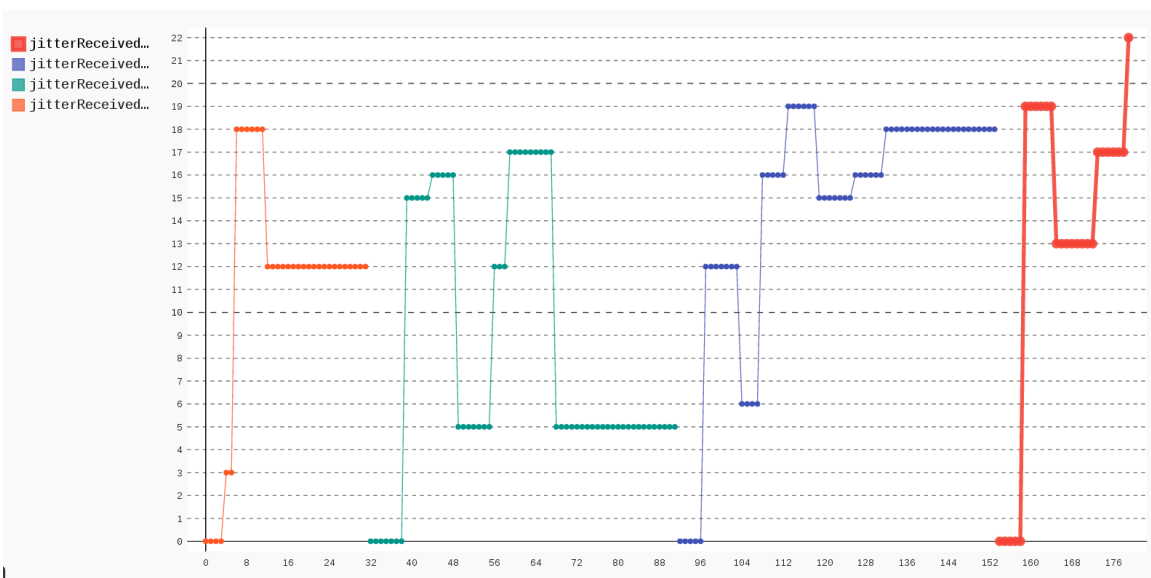
La calidad de servicio depende de: capacidad del hardware, velocidad de transmisión de datos, eficiencia del algoritmo, etc. Para el análisis de la efectividad de la plataforma se ha recurrido a analizar registros de CE003, pretendiendo entender el comportamiento de las variaciones del retraso, jitter, memoria temporal de jitter y de los paquetes perdidos, tanto para audio como para video.

##### 4.9.1. Audio

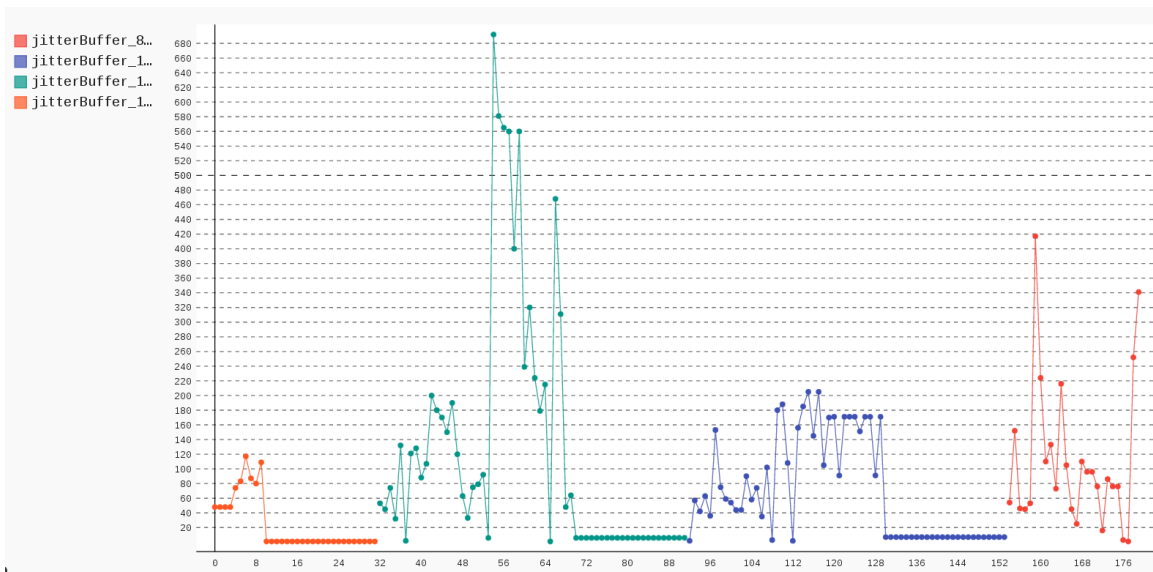
CE003 tiene 3 desconexiones, por lo que en la (figura 43) se evidencian 4 conexiones, marcadas con distintos colores. Los lapsos de desconexión corresponden a los espacios vacíos entre 2 desconexiones. Como se puede visualizar en la segunda conexión existe un pico en el segundo 54, donde el retardo se dispara a 560, este mismo salto se refleja en la (figura 44) y la (figura 45), porque si existe un retardo repentino, el jitter debe incrementa de igual manera, así como también la información almacenada en jitterBuffer (figura 45).



**Figura 43 Variación del Retraso de audio en CE003**

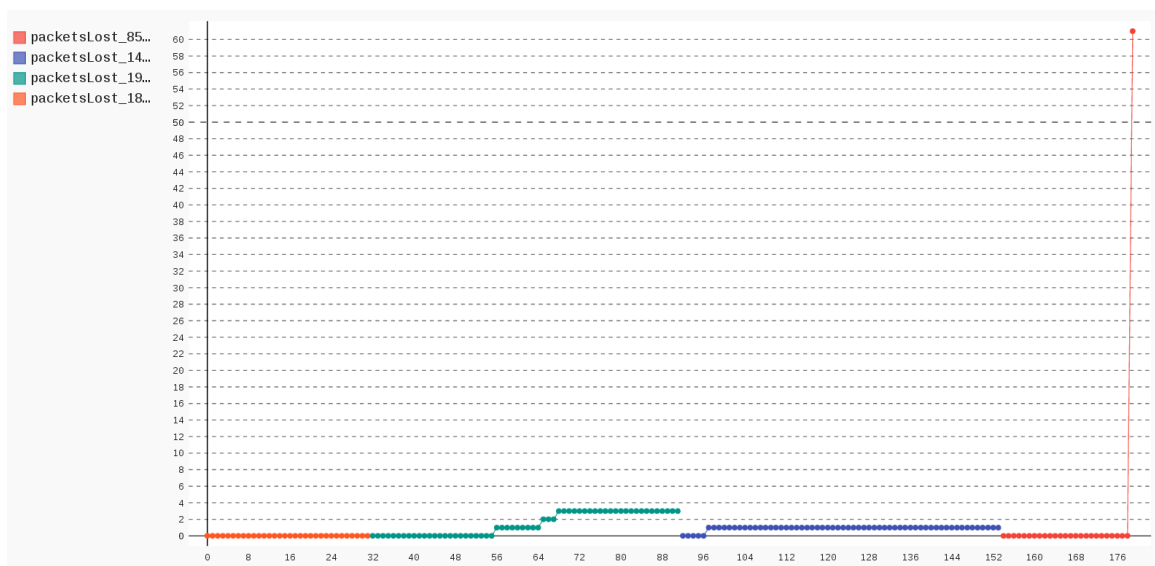


**Figura 44 Variación del Jitter Recibido de audio en CE003**



**Figura 45 Variación del almacenamiento JitterBuffer de audio en CE003**

El pico generado en el segundo 54 también puede resultar con pérdida de paquetes en el caso de que la memoria jitterBuffer esté saturada. En éste caso, debido a los retrasos de la Red, se evidencian pérdida de 2, 3 y 4 paquetes a partir del segundo 56 (figura 46). Por otro lado también se pueden ver pérdidas en el segundo 176, cuando la llamada ha finalizado, los mismo que no superan el mínimo de 1% establecido por recomendación.

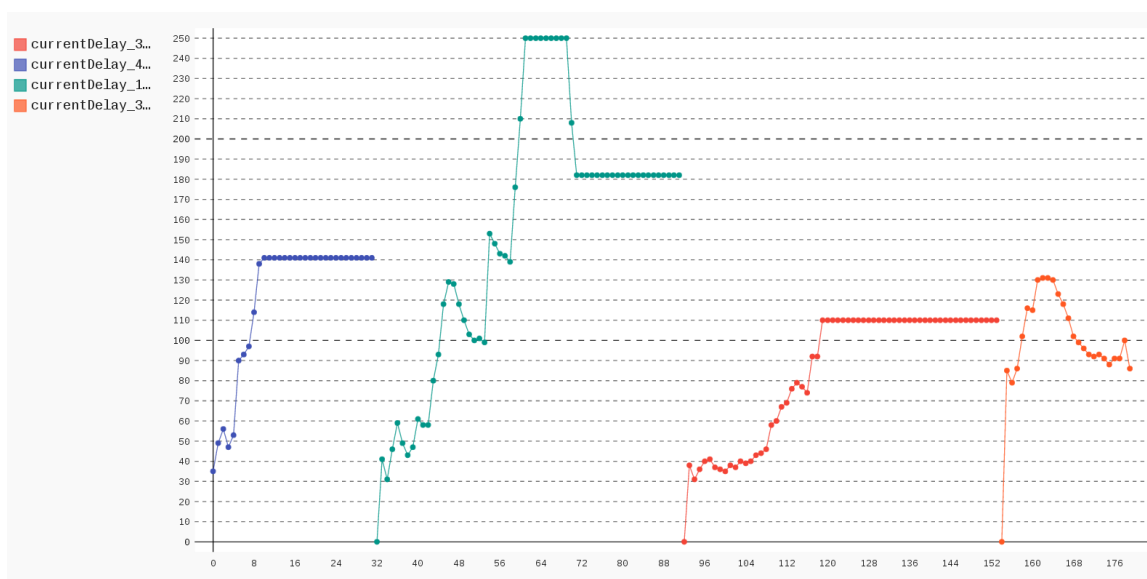


**Figura 46 Variación de la pérdida de paquetes de audio en CE003**

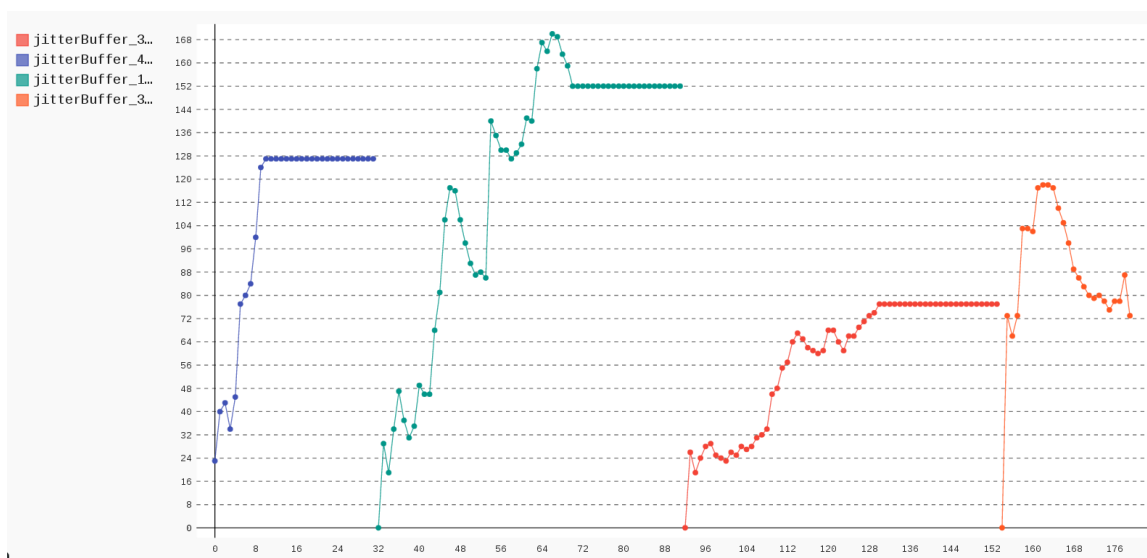


### 4.9.2. Video

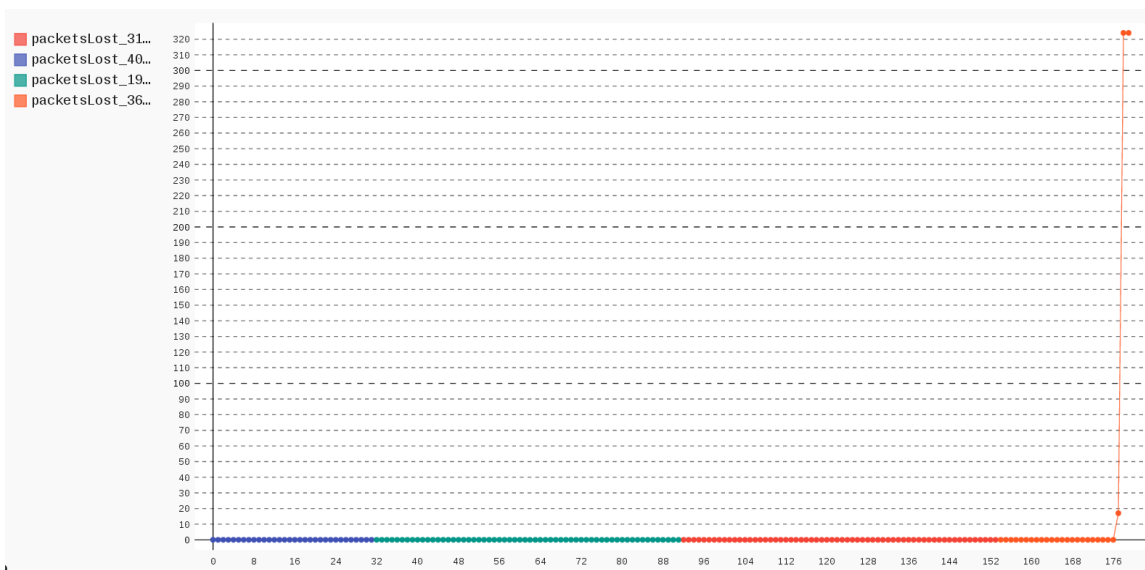
En cuanto a video, el retraso no supera los 400ms de la recomendación, llega a 260 en la (figura 47) sin resultar con pérdidas de paquetes (figura 49). Sin embargo, se puede visualizar que este retraso si incrementa el jitter (figura 48) y el almacenamiento en jitterBuffer a partir del segundo 61, al igual que el audio. Con esto se puede concluir que la velocidad de transmisión de datos ha disminuido en éste tiempo, posiblemente por congestiónamiento.



**Figura 47 Variación de la Retraso de video en CE003**



**Figura 48 Variación del almacenamiento de jitterBuffer de video en CE003**



**Figura 49 Variación de paquetes perdidos de video en CE003**

#### 4.10. Experiencia de usuarios en la plataforma

La calidad de experiencia se ha evaluado directamente a través de la opinión de una muestra de usuarios mediante la métrica de evaluación directa MOS y un método sin comparación. El audio y video se transportan por canales distintos en WebRTC, codificando su grabación en formatos WebM para Video y Waveform Audio File Format (WAV) para voz.

La recomendación UIT-T P.800 (UIT, 96) es una guía para los administradores que desean realizar pruebas de calidad de transmisión, la cual define los métodos de determinación subjetiva de la calidad de transmisión y proporciona una descripción de métodos recomendados para las pruebas de opinión sobre la escucha, conversación.

Existen muchos métodos para determinar la calidad del audio, sin embargo se necesita de una evaluación que no compare una señal con otra de referencia, donde el usuario pueda evaluar su opinión sobre la escucha, razón por la cual el método ideal es el de índices de categorías absolutas Absolute Category Rating (ACR), donde el audio se evalúa con valores entre 1 al 5, donde 5 es “Excelente” y 1 “Malo”.

Por otro lado, la evaluación de la calidad del video se basa en los métodos de evaluación subjetiva propuestos por la ITU-T P.910 (ITU, 2008) para aplicaciones multimedia. Donde el método ACR es el indicado para emitir un juicio de categorías,

puesto que se presenta una sola vez la secuencia de video en tiempo de real de la videoconferencia.

Los valores de la calidad mediante MOS, tanto como para audio y video se dan gracias al promedio de los valores ACR de las pruebas realizadas a una población de usuarios que se especifica a mayor detalle en el apartado de la descripción del entorno de las pruebas subjetivas.

#### **4.10.1. Participantes de las pruebas**

En base a la revisión de literatura de las publicaciones (Liu, 2012), (Zhang, 2013) (Song, 2014) (Ickin, 2015) (Zeng, 2016) ,que tratan sobre calidad de experiencia en servicios de streaming de video y video 3D trabajan en sus evaluaciones con 21, 22, 30, 30 y 23 sujetos de prueba, por lo que para esta evaluación se ha tomado la media de los participantes de los trabajos citados, dando como resultado una necesidad de invitar a 25 usuarios no expertos.

#### **4.10.2. Descripción del entorno y de las pruebas subjetivas**

Se han realizado pruebas con la participación entre 2 y 3 usuarios por video llamada, donde 25 interlocutores se han ubicado dentro de una sala de 16 m<sup>3</sup> aproximadamente. Para evitar el ruido del medio en la grabación y en la reproducción del audio se ha optado por el uso de audífonos y micrófonos, con un volumen del 40% del sistema. Mientras que para una buena apreciación del video se ha limitado la entrada de luz solar.

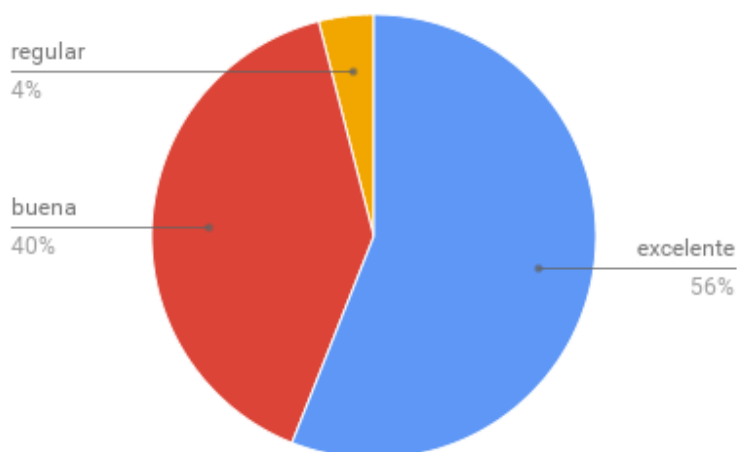
Cada usuario ha tenido que evaluar para el audio la calidad de escucha y su escala de esfuerzo, y para el video la nitidez de la imagen, tanto como para el tiempo en que se da la videoconferencia, así como también para la información grabada y recuperada del tiempo de desconexión, donde se apaga manualmente de la red inalámbrica un lapso de 20 segundos aproximadamente, durante la prueba que teóricamente tiene una duración de 3 minutos.

#### 4.11. Resultados de las pruebas de calidad de experiencia

##### 4.11.1. Calidad de escucha

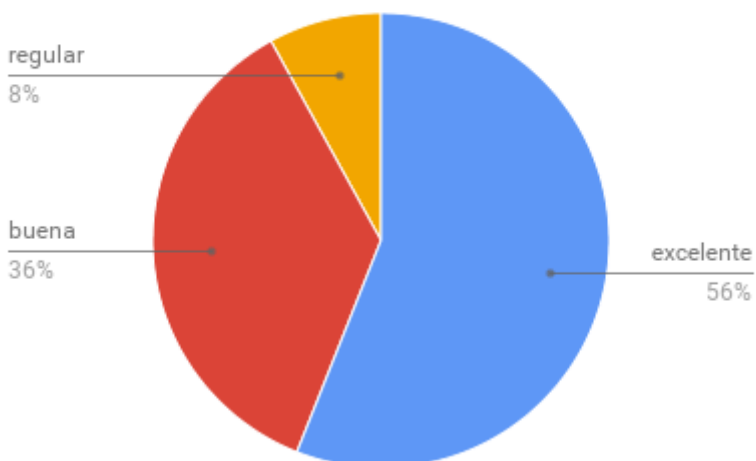
Calidad de escucha o de señal vocal describe la nitidez de la voz que es codificada, los valores ACR que dispone el usuario en la encuesta se describen de la siguiente manera: 5 como “Excelente”, 4 “Buena”, 3 “Regular”, 2 “Mediocre” y 1 igual a “Mala”. La distribución de éstos valores se han dado como lo describe la (figura 50) para videoconferencia y la (figura 51) para la información recuperada del tiempo de desconexión.

Recuento de Calidad de escucha en Videoconferencia



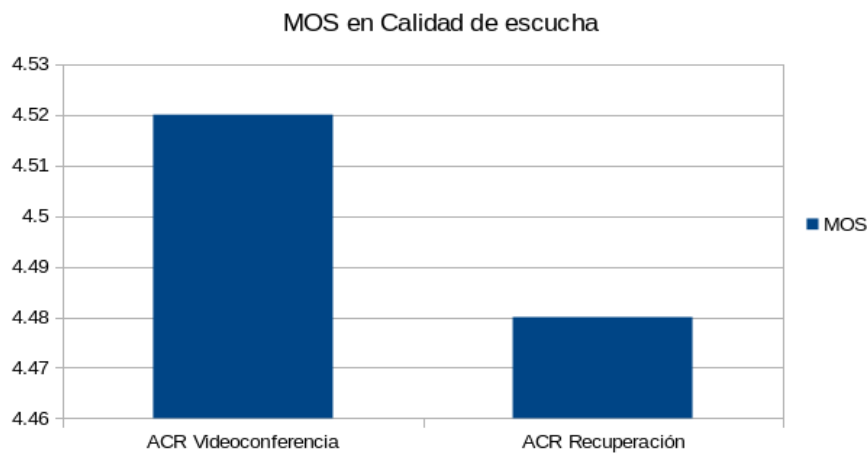
**Figura 50** Calidad de escucha en videoconferencia

Recuento de Calidad de Escucha en Información recuperada



**Figura 51** Calidad de escucha en recuperación

Con las calificaciones de los usuarios, se procede a encontrar el resultado para MOS mediante el cálculo de la media de los valores ACR, dando como resultados las cantidades de 4,52 para la calidad de escucha en la videoconferencia y 4,48 para la información recuperada como lo ilustra la (figura 52), resultados que se interpretan como “Excelente” y “Buena” en MOS.

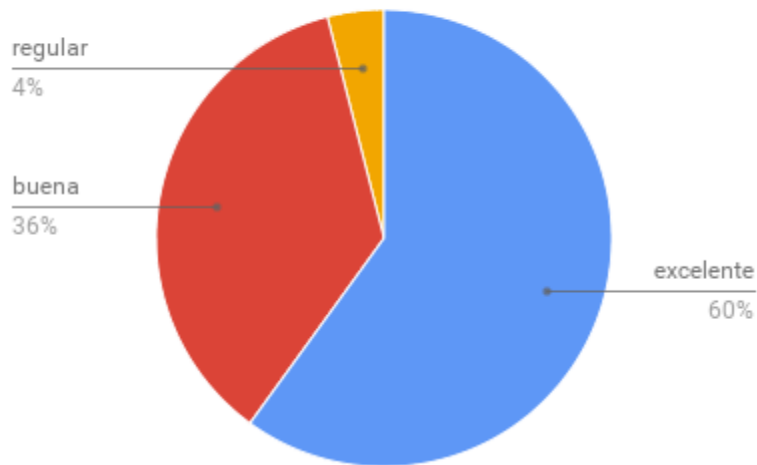


**Figura 52 MOS de la calidad de escucha**

#### 4.11.2. Esfuerzo de escucha

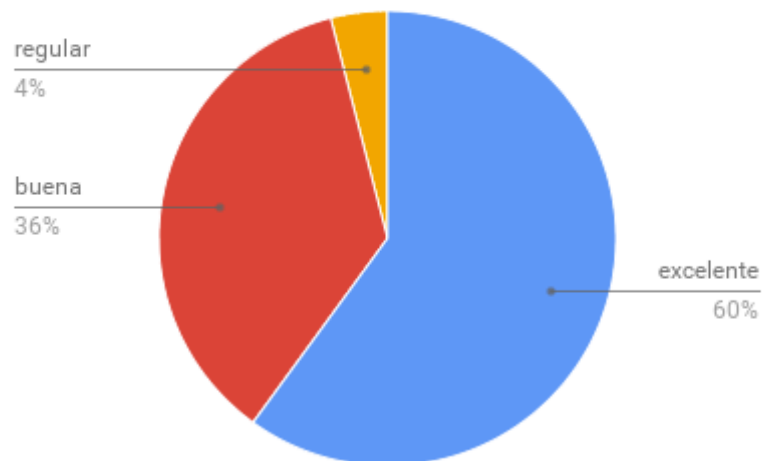
La descripción del significado de las escalas de esfuerzo de escucha es importante mencionar, puesto que si no son aclaradas podría causar confusiones a los usuarios calificadores. La calidad de esfuerzo de escucha se describe como el esfuerzo necesario para comprender el significado de las frases, la interpretación de los valores ACR vienen dados de la siguiente manera: 5 como “Excelente; Audición Perfecta”, 4 “Buena; cierta atención es necesaria”, 3 “Regular; Esfuerzo moderado”, 2 “Mediocre; Esfuerzo considerable” y 1 como “Mala; Significado incomprensible”. La distribución de éstos valores se han dado como lo describe la (figura 53) para la videoconferencia y la (figura 54) para la información recuperada del tiempo de desconexión.

Recuento de Esfuerzo de escucha en Videoconferencia



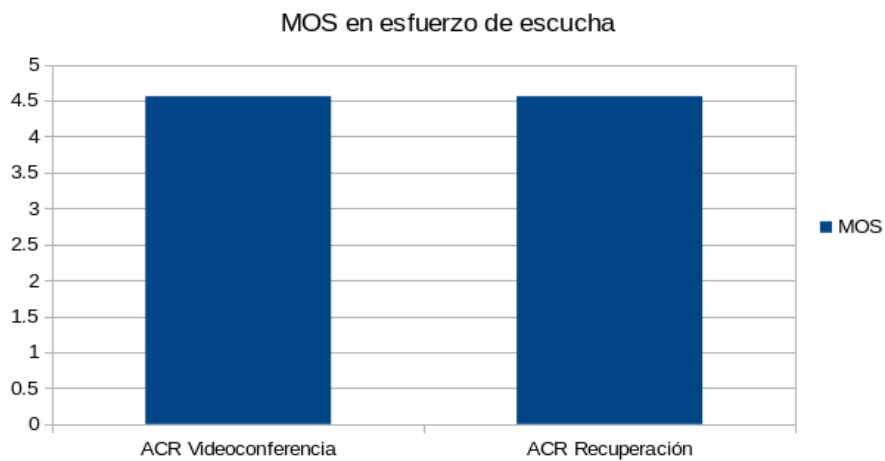
**Figura 53 Esfuerzo de escucha en videoconferencia**

Recuento de Esfuerzo de escucha en Información recuperada



**Figura 54 Esfuerzo de escucha en recuperación**

Los valores de MOS para el esfuerzo de escucha mediante el cálculo de la media de ACRs, dan como resultados 4,56 tanto para la calidad de escucha en la videoconferencia, como para la información recuperada que ilustra la (figura 55), que se interpretan como “Excelente” para los dos casos y demuestran que calidad de los elementos de WebRTC tienen muy buena acogida.

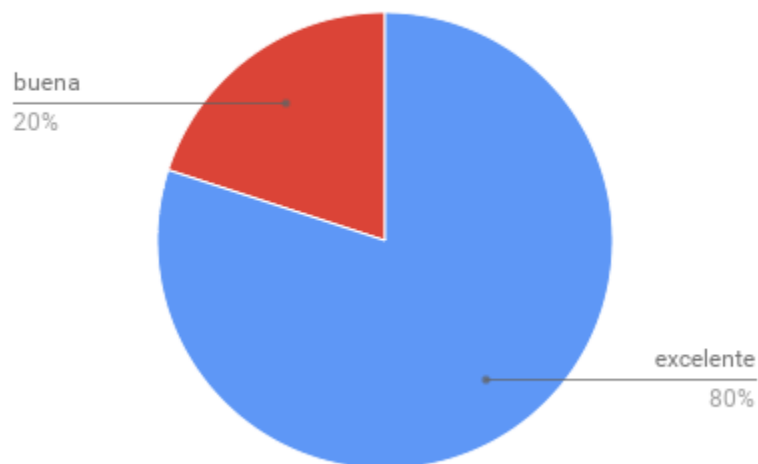


**Figura 55 MOS de la calidad de escucha**

#### 4.11.3. Nitidez de la imagen

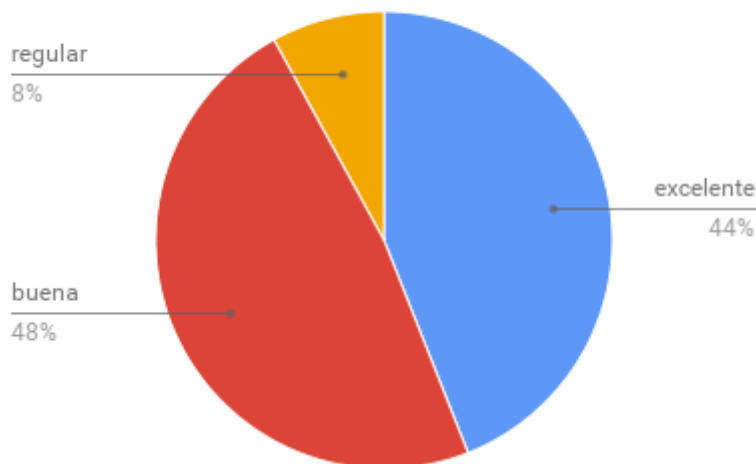
La descripción de los valores de la escala de la calidad de la imagen del video no requiere mayor explicación, las escalas ACR son: 5 a la mejor “Excelente” y 1 para la más baja “Mala”. En las evaluaciones realizadas se han obtenido los resultados que se visualizan en la (figura 56) para la videoconferencia y la (figura 57) para la información recuperada del tiempo de desconexión.

Recuento de Calidad de Video en Videoconferencia



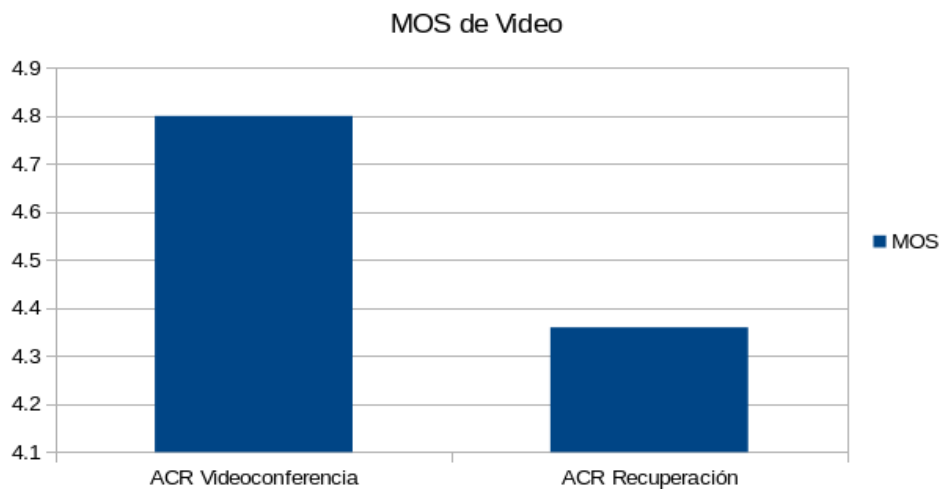
**Figura 56 Calidad de video en videoconferencia**

### Recuento de Calidad de Video en Información recuperada



**Figura 57 Calidad de video en recuperación**

Los valores de MOS para la calidad del video mediante el cálculo de la media de ACRs, dan como resultados los números 4,8 para la videoconferencia y 4,36 para para la información recuperada (figura 58). Los valores se interpretan como “Excelentes” y “Buenos” respectivamente, pero tienen una contradicción en cuanto a la calidad de codecs de video que se utilizan, esto se debe a que a pesar de que la información recuperada utiliza VP9, ésta calidad de video ha sido reducida desde la resolución de la grabación del contenido de la cámara, por motivos de que el transporte de la información recuperada no sea tan larga y se pueda recuperar oportunamente.



**Figura 58 MOS de la calidad de video**



#### 4.11.1. Máximo de usuario por sala de videoconferencia

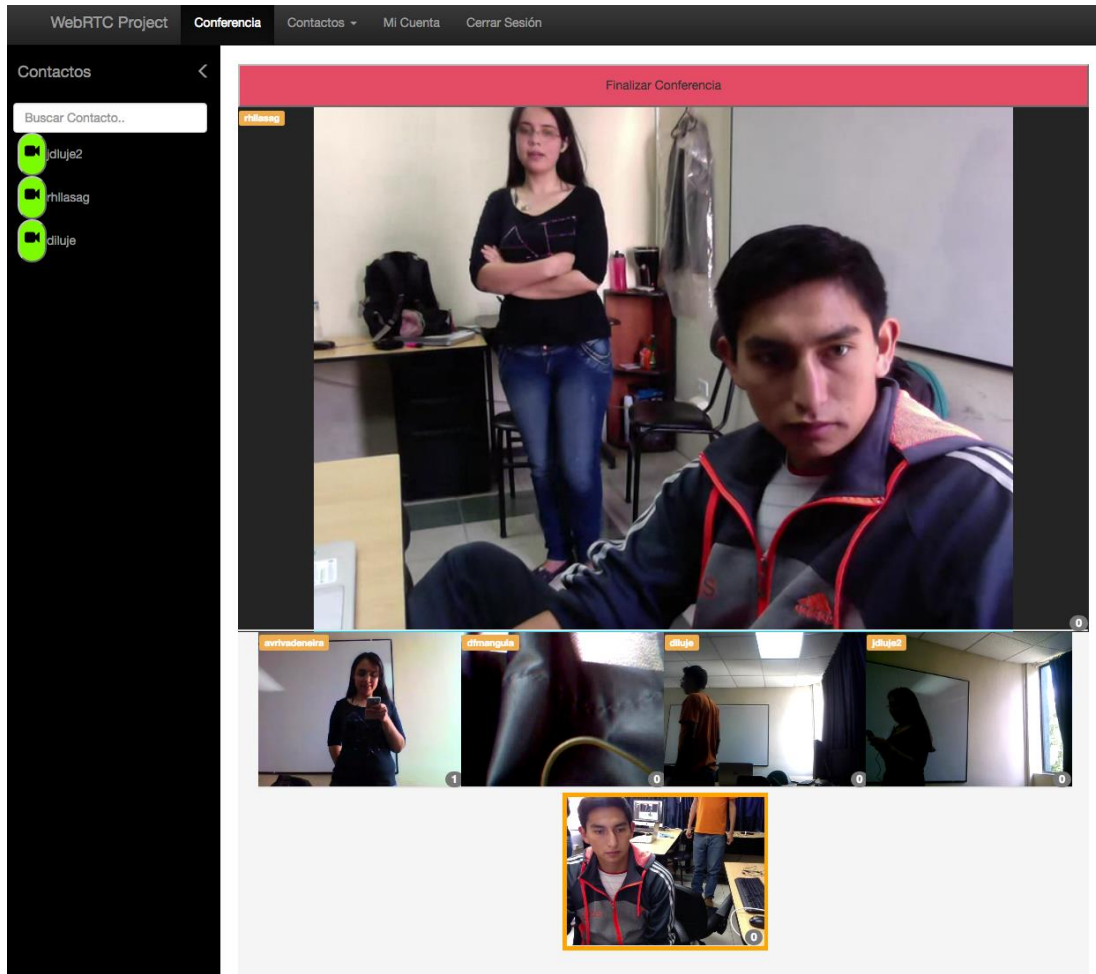
Pese a que el servicio de videoconferencia múltiple está orientado para una participación de  $n$  usuarios en una misma sala común, se ha llegado a concluir que el número máximo de usuarios que puede soportar la red en la que se han realizado estas pruebas es de cinco. Puesto que en la (tabla 38), se recogen la información de 5 canales de comunicación de una sala de 6 interlocutores (figura 59), donde transmiten un total de 472146.91 bytes transmitidos por segundo de video y 18891.90 de audio al sexto integrante, datos que al sumar y transformar a Megabits con la ayuda de la (ecuación 12) da un resultado de 3.746 Mb, información que al parecer empieza a saturar la red y a delimitar el escenario ideal, puesto que el retraso para audio y video superan los 400 milisegundos que recomienda la UTI-T G.114.

$$y = \frac{x \text{ byte} * Mb}{(1024 * 1024)bits} \times \frac{8bits}{byte}$$

#### Ecuación 13 bytes a Megabits

**Tabla 38**  
**Estadísticas de Audio y Video en videoconferencia con 6 usuarios**

	Recibido						Enviado		
	Video			Audio			Video	Audio	
	Jitter Recibido	Retraso	Jitter Recibido	Retraso	Jitter Recibido	Retraso	BytesPerSec	Jitter Received	BytesPerSec
# Prueba	Media	Media	Media	Media	Media	Media	Media	Media	Media
log_user1.json	434,94	43639,89	444,51	415,54	24,75	4450,55	109229,83	22,71	4798,76
log_user2.json	307,59	4715,60	214,23	333,18	25,91	1433,41	98604,50	15,89	4934,04
log_user3.json	273,00	67572,86	258,68	553,38	27,20	4611,40	47906,19	23,93	4923,95
log_user4.json	128,52	224612,50	84,75	158,22	8,96	5206,55	101708,14	14,05	5004,30
log_user5.json	101,31	131606,06	61,50	113,07	7,81	3189,99	94551,82	10,47	5021,53



**Figura 59** MOS de la calidad de video

## CAPÍTULO V

### CONCLUSIONES Y TRABAJOS FUTUROS

#### 5.1. Conclusiones

- Se acepta la hipótesis nula al lograr reducir las pérdidas de información del tiempo de desconexión a un promedio de 2 segundos, lapso formado por el instante en que se desconecta el hardware y el software es notificado, tomando en cuenta los retrasos de red y el tiempo de espera establecido.
- Pese a que el algoritmo del mecanismo de control de desconexión está diseñado para una participación de varios usuarios, las limitaciones del canal de comunicaciones y del hardware han limitado su funcionamiento óptimo para 5 usuarios en una sala de videoconferencia común.
- Se ha elegido trabajar con WebRTC por ser de código libre, multiplataforma, no depende de la instalación de componentes adicionales y utiliza codecs de video orientados a la transferencia por la web, además de usar protocolos seguros.
- La comunicación de WebRTC requiere de un servidor intermediario para establecerla, por lo que se ha diseñado la plataforma de videoconferencia múltiple sobre una arquitectura P2P híbrida y bajo los patrones MVC y Proxy.
- Los usuarios demuestran descontento al perder la continuidad de la comunicación; Malestar al iniciar manualmente una nueva llamada; Pérdidas de información durante el tiempo de desconexión.
- El patrón Proxy es utilizado para censar continuamente el estado de la conexión de cada punto con el servidor, para que en el caso de que se diera una desconexión se pueda restablecer ésta comunicación, siendo transparente para los usuarios.
- La Arquitectura de la Plataforma de videoconferencia múltiple dividida por capas ha sido totalmente adaptable a la incorporación de las funcionalidades del mecanismo de control de desconexiones, las cuales se basaron en historias de usuario de la metodología XP.
- La implementación del mecanismo de control de desconexión no afecta significativamente al rendimiento de WebRTC, puesto que las pruebas de QoS demuestran que los valores para retardo, jitter e información perdida se encuentran dentro de los parámetros recomendados.
- Las pruebas de QoE reflejaron que el audio y video de la videoconferencia e información recuperada recibieron una calificación promedio (MOS) de “Bueno” y “Excelente” desde la perspectiva subjetiva de los usuarios no expertos.

## 5.2. Trabajos futuros

- Se propone para trabajos futuros usar el modelo de sistemas distribuidos P2P Supernodos para minimizar la dependencia de un servidor controlador de establecimiento de comunicaciones, además de tener la capacidad de que un usuario pueda participar en varias salas de videoconferencia simultáneamente.
- Realizar un sistema de predicción que analizando las estadísticas de WebRTC permita a los puntos participantes de una videoconferencia múltiple ser proactivos y anticiparse a una posible interrupción de la red para eliminar completamente la información perdida durante la desconexión.
- Utilizar el protocolo P2PSP para la transmisión de segmentos entre los participantes de la videoconferencia, minimizando así la carga en los participantes con menos recursos.

## BIBLIOGRAFÍA

- 3CX. (23 de 5 de 2013). *Por qué las videoconferencias cierran mas ventas?* Obtenido de 3CX:  
<http://www.3cx.es/blog/videoconferencias-cierran-mas-ventas/>
- 3CX. (23 de 5 de 2016). *3CX-WebRTC*. Obtenido de <http://www.3cx.es/webrtc/>
- ALEGSA. (2016). *Diccionario de la Informática y Tecnología*. (ALEGSA, Productor)  
 Recuperado el 27 de 9 de 2016, de Framework:  
<http://www.alegsa.com.ar/Dic/framework%20para%20aplicaciones%20web.php>
- Amin, R. H. (2016). Video QoS/QoE over IEEE802.11n/ac: A Contemporary Survey.  
 (ProQuest, Ed.) *ProQuest* (10142999).
- Amir, Y. (2013). Peer-to-peer architectures for massively multiplayer online games: A Survey.  
*ACM*.
- Arellano, A., & Fernández, J. (06 de 2014). *Repositorio ESPE*. Obtenido de  
<http://repositorio.espe.edu.ec/bitstream/21000/8882/1/T-ESPE-048109.pdf>
- Atelin, P. (2007). *TCP/IP y protocolos de Internet*.
- Berners-Lee, T. (05 de 1996). *RFC 1945*. (RFC, Productor, & RFC) Obtenido de Hypertext  
 Transfer Protocol: <https://www.rfc-editor.org/info/rfc1945>
- Bonini, J. (2012). *Cumplir con las expectativas del cliente en la entrega de Vídeo sobre IP*.  
 IEEE.
- Bootstrap. (19 de 8 de 2011). *About*. (Bootstrap, Productor) Recuperado el 27 de 11 de 2016,  
 de <http://getbootstrap.com/about/>
- Candelas, F. (2009). *Protocolos de Transporte TCPy UDP*. Universidad de Alicante, Física,  
 Ingeniería en Sistemas y Teoría de la Señal.
- Carlos Genis Triana, A. R. (2004). *Evaluación de una versión paralela para el Codec  
 H.264/AVC*. Instituto Tecnológico de Veracruz, Sistemas y Computación. Veracruz:  
 Instituto Tecnológico de Veracruz.

Carrera, A. (2015). *DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA DE ESTUDIO*.

Santiago de Chile: Universidad de Chile.

Castejón, J. S. (2004). Arquitectura y diseño de sistemas web modernos. *Ingeniería de Software del CIIRM*.

CISCO. (18 de 5 de 2008). *IP: Servicios de direccionamiento IP*. (CISCO, Productor)

Recuperado el 30 de 9 de 2016, de Preguntas frecuentes sobre NAT:

[http://www.cisco.com/cisco/web/support/LA/7/76/76169\\_nat-faq.html#qa1](http://www.cisco.com/cisco/web/support/LA/7/76/76169_nat-faq.html#qa1)

CISCO. (2 de 2 de 2006). *Understanding Jitter in Packet Voice Networks (Cisco IOS*

*Platforms)*. (Cisco, Productor) Recuperado el 14 de 9 de 2016, de

<http://www.cisco.com/c/en/us/support/docs/voice/voice-quality/18902-jitter-packet-voice.html>

Clark, A., & Clark, E. (2004). Diccionario Inglés a Español de Computación e Internet:

English-to-Spanish Computer and Internet Dictionary.

Clark, J. (2013). SoK: SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements. *IEEE*.

Cueto Díaz. (2004). *Estudio e Implementación de un Cluster Clase*. Pontificia Universidad

Católica de Valparaíso (PUCV). Facultad de Ingeniería. Escuela de Ingeniería Eléctrica (EIE), Chile.

Dávila Neuman, G. (2006). *El razonamiento inductivo y deductivo dentro del proceso*

*investigativo en ciencias experimentales y*. Caracas: Laurus.

Díaz, A., Merino, P., Panizo, L., & Recio, Á. M. (s.f.). Un estudio práctico del rendimiento del servicio de Streaming de Video sobre redes móviles GPRS/UMTS. *Universidad de Málaga*.

Dutton, S. (4 de 11 de 2013). *html5rocks*. Recuperado el 30 de 9 de 2016, de WebRTC in the real world: STUN, TURN and signaling:

<https://www.html5rocks.com/en/tutorials/webrtc/infrastructure/#after-signaling-using-ice-to-cope-with-nats-and-firewalls>

Fernández, Y., & Díaz, Y. (2012). Patrón Modelo Vista Controlador. *Telemática*, 47-57.

Gatica. (2005). *Transversalidad en Nat/Firewall*. Universidad Católica de Temuco.

González, L. (2009). *MÉTODO PARA GENERAR CASOS DE PRUEBA FUNCIONAL EN EL DESARROLLO DE SOFTWARE*. Universidad de Antioquia, Ingeniería en Sistemas, Antioquia.

Google Developers. (2016). *Google Developers*. (S. Dutton, Productor) Obtenido de VP9 is now available in WebRTC: <https://developers.google.com/web/updates/2016/01/vp9-webrtc>

Gourley, D. (2002). *HTTP: The Definitive Guide: The Definitive Guide*.

Granda. (2015). *Overlay network based on WebRTC for interactive multimedia communications*.

Grigorik, I. (2013). *High Performance Browser Networking*. (O. R. Media, Productor, & O'Reilly Media) Recuperado el 27 de 9 de 2016, de High Performance Browser Networking: <https://hpbn.co/webrtc/>

Hammad, K. (26 de 1 de 2016). Analytical Approximation of Packet Delay Jitter in Simple Queues. (IEEE, Ed.) *IEEE*.

Huidrobo, J. M., & Roldan, D. (2003). *Integración de voz y datos: call centers, tecnología y aplicaciones* (1 ed.). Madrid, España: S.A. MCGRAW-HILL / INTERAMERICANA DE ESPAÑA.

IANA. (1 de 10 de 2016). *IANA*. (iana, Productor) Recuperado el 27 de 11 de 2016, de <http://www.iana.org/>

Ickin, S. (2015). VLQoE: Video QoE instrumentation on the smartphone. *Springer*.

- IETF. (10 de 2008). *Internet Engineering Task Force*. (RFC, Productor, & RFC) Recuperado el 29 de 10 de 2016, de Transport Layer Security TLS version 1.2:  
<https://tools.ietf.org/html/rfc5246>
- IETF. (12 de 2011). *Internet Engineering Task Force* . (R. 6455, Productor) Recuperado el 22 de 9 de 2016, de The WebSocket Protocol: <https://tools.ietf.org/html/rfc6455>
- IETF. (1992). *The INternet Engineering Task Force*. (IETF, Productor) Recuperado el 27 de 11 de 2016, de <https://www.ietf.org/>
- IETF. (2 de 2013). *Internet Engineering Task Force*. (IEFT, Productor) Obtenido de SDP for WebRTC: <https://tools.ietf.org/id/draft-nandakumar-rtcweb-sdp-01.html>
- IETF. (3 de 2016). *Internet Engineering Task Force*. (RFC, Productor, & RFC) Recuperado el 30 de 9 de 2016, de WebRTC Video Procesing and Codec Requirements:  
<https://tools.ietf.org/html/rfc7742>
- IETF. (5 de 1992). *Internet Engineering Task Force*. (RFC, Productor, & RFC) Recuperado el 29 de 09 de 2016, de TCP Extensions for high performance:  
<https://tools.ietf.org/html/rfc1323>
- IETF. (7 de 2006). *Internet Engineering Task Force*. (RFC, Productor, & RFC) Obtenido de SDP: Session Description Protocol: <https://tools.ietf.org/html/rfc4566>
- IETF. (8 de 2011). *Internet Engineering Task Force*. (IETF, Productor, & IETF) Recuperado el 7 de 10 de 2016, de Requisitos para un códec de audio de Internet:  
<https://tools.ietf.org/html/rfc6366>
- IETF. (9 de 2012). *Internet Engineering Task Force*. (IETF, Productor, & IETF) Recuperado el 7 de 10 de 2016, de Definición del Opus Audio Codec:  
<https://tools.ietf.org/html/rfc6716>



- ITU International Telecommunication Union. (07 de 05 de 2007). *Definition of uality of Eperience*. Recuperado el 21 de 9 de 2016, de <https://www.itu.int/md/T05-FG.IPTV-IL-0050/en>
- ITU. (2008). *Recommendation ITU-T P.910*. International Telecommunication Union, Telecommunication Standarization Sector of ITU. ITU.
- ITU. (2016). *ITU*. Recuperado el 27 de 11 de 2016, de <http://www.itu.int/es/about/Pages/default.aspx>
- K Dictionaries Ltd. (2013). *Umbral*. Recuperado el 15 de 9 de 2016, de <http://es.thefreedictionary.com/>
- Karimov, I. (2010). Reconnection of the Pre-assigning Peers in the Seamless Video Streaming. *IEEE* , 2.
- Lewis, C., & Pickavance, S. (26 de 5 de 2006). *Implementing Quality of Service Over Cisco MPLS VPNs*. Recuperado el 12 de 11 de 2016, de <http://www.ciscopress.com/articles/article.asp?p=471096&seqNum=6>
- Liu, Y. (2012). QoE-Oriented 3D Video Transcoding for Mobile Streaming. *ACM* .
- Lorenzo, K., & Anh, L. (2012). Microfundido: vídeo streaming de cooperación en los teléfonos inteligentes. *ACM* .
- Marcillo, D. (2015). *Control de interrupciones de video streaming móvil en arquitecturas Android usando técnicas de realidad aumentada y WebRTC*. Univesidad de Las Palmas de Gran Canaria, Sistemas e Informática, Las Palmas de Gran Canaria.
- Medina, A. C. (12 de 2003). La videoconferencia: conceptualización, elementos y uso educativo. *Etic@net* .
- Microsoft Developer. (2016). *Stream IO Stream*. Recuperado el 14 de 9 de 2016, de <https://social.msdn.microsoft.com/Search/en-US?query=stream&pgArea=header&emptyWatermark=true&ac=4>

- Microsoft. (2012). *Introducción a la Calidad de servicio (QoS)*.
- Millán, R. (2006). Características de P2P. En *Distribución de libros electrónicos en redes P2P*. ACTA y CEDRO.
- Montenegro, P., & Mora, V. (2007). *Análisis y Evaluación para la selección de codecs de VoIP*. Universidad de Azuay, Facultad de Administración. Cuenca: Universidad de Azuay.
- Mozilla Foundation. (2016). *HTML5*. <https://developer.mozilla.org/es/docs/HTML/HTML5>.
- Mozilla. (12 de 05 de 2016). *Mozilla Developer Network* . Recuperado el 19 de 9 de 2016, de HTML5: <https://developer.mozilla.org/es/docs/HTML/HTML5>
- Mozilla. (20 de 12 de 2015). *WebSockets*. (C. Bustios, Productor, & Mozilla) Recuperado el 18 de 09 de 2016, de Mozilla developer Network: <https://developer.mozilla.org/es/docs/WebSockets-840092-dup>
- Mukherjee, D., & Bankoski, J. (2013). The latest open-source video codec VP9 – An overview and preliminary results. (G. Inc, Ed.) *IEEE* .
- Murazzo, M., Rodriguez, N., Vergara, R., Carrizo, F., Gonzalez, F., & Grosso, E. (2013). *Administración de QoS en ambientes de redes de servicios convergentes*.
- Murillo, J. (2008). *PostgradoUNE*. Obtenido de <http://www.postgradoune.edu.pe/documentos/Experimental.pdf>
- NTT Communications. (28 de 07 de 2015). *A Study of WebRTC Security* . Recuperado el 26 de 9 de 2016, de <http://webrtc-security.github.io/>
- Opticom. (2016). *PEV – Perceptual Evaluation of Video Quality*. Recuperado el 11 de 12 de 2016, de <http://www.opticom.de/technology/technology.php>
- Orenstein, D. (2010). *Application Programming Interface*.
- Patrick Le Callet, S. M. (3 de 2013). Qualinet White Paper on Definitions of Quality of Experience. *QUALINET* .

- Potencier, F. (2009). *El tutorial Jobeet*. eBook.
- Python Org. (2001). *General Python FAQ*. Recuperado el 27 de 11 de 2016, de General Python  
FAQ: <https://docs.python.org/3/faq/general.html#what-is-python>
- RAE. (2016). *Diccionario de la Lengua Española*.
- Real Time Communications. (2016). *What is Real Time Communications?* (R. Ramsey, Productor) Recuperado el 27 de 11 de 2016, de  
<http://www.realtimemunicationsworld.com/topics/realtimemunicationsworld/articles/376916-what-real-time-communications.htm>
- REGO. (2006). *Balanceamento de carga em clusters de alto desempenho: uma extensão para a LAM/MPI*. Universidade Estadual de Maringá.
- Rescorla, E. (2000). *SSL and TLS: Designing and Building Secure Systems: Building and Designing Secure Systems* (1 ed.). ADDISON WESLEY LONGMAN INC DIV PEARSON SUITE 300.
- Romo. (2003). "*Procesamiento Paralelo*". *Universidad Internacional del Ecuador*. Obtenido de <http://www.internacional.edu.ec/academica/informatica/creatividad/uide-bits/uide-bits-05->
- Sabino, C. (02 de 2008). *El proceso de Investigación*. Obtenido de [https://metodoinvestigacion.files.wordpress.com/2008/02/el-proceso-de-investigacion\\_carlos-sabino.pdf](https://metodoinvestigacion.files.wordpress.com/2008/02/el-proceso-de-investigacion_carlos-sabino.pdf)
- Sánchez, M. (2012). Javascript. En M. S. Maza, *Javascript*.
- Sciara, D. R. (2004). Fundamentos de Video Streaming. *Universidad de la República* .
- Skirelis, J., & Serackis, A. (2015). Prediction of the Real-Time Video Streaming Performance based on the Peer Connection Statistics. *IEEE* , 4.
- Song, J. (2014). QoE Space based QoE Adaptation Algorithm for SVC-P2P Video Streaming Systems. *IEEE* .

- Steinmetz, R., & Wehrle, K. (2005). What Is This “Peer-to-Peer” About? *Springer Berlin Heidelberg* , 9-16.
- Tavakoli, S. (2015). *Subjective QoE Analysis of HTTP Adaptive Streaming Applications*. Madrid: Universidad Politécnica de Madrid.
- Thakur, A. (2012). *Video Conferencing*. Recuperado el 12 de 9 de 2016, de <http://www.engineersgarage.com/articles/video-conferencing?page=2>
- Transparency Market Research. (23 de 05 de 2016). Obtenido de Video Conferencing Market to Reach US\$7.85 bn by 2023, Driven by Globalization of Enterprises.: <http://www.transparencymarketresearch.com/pressrelease/video-conferencing-market.htm>
- Uca. (2016). *Formatos de URL*. Recuperado el 27 de 11 de 2016, de <http://www2.uca.es/serv/sc/manual-html/url.htm>
- UIT. (2003). *Recomendación UIT-T G.114 Tiempo de transmisión en un sentido*. Union Internacional de Telecomunicaciones.
- UIT. (96). *Recomendación UIT-T P.800*. Unión Internacional de Telecomunicaciones, Sector de Normalización de las Telecomunicaciones de la UIT. UIT.
- W3C. (2016). *W3C España*. Recuperado el 27 de 11 de 2016, de <http://www.w3c.es/ayuda/>
- W3C. (2016). *W3C*. Obtenido de <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>
- w3cschools. (2016). *w3cschools*. Recuperado el 28 de 10 de 2016, de local storage HTML5 : [http://www.w3schools.com/html/html5\\_webstorage.asp](http://www.w3schools.com/html/html5_webstorage.asp)
- WebM Project. (2010). *WebM Project*. Recuperado el 27 de 11 de 2016, de <http://www.webmproject.org/about/>
- WebRTC Glossary. (5 de 10 de 2014). *WebRTC Glossary*. (W. Glossary, Productor) Recuperado el 26 de 9 de 2016, de RTP: <https://webrtcglossary.com/rtp/>

- WebRTC Glossary. (5 de 9 de 2014). *WebRTC Glossary*. (W. Glossary, Productor) Recuperado el 26 de 10 de 2016, de SRTP: <https://webrtcglossary.com/srtp/>
- Wells, D. (8 de Octubre de 2013). *Extreme Programming*. Obtenido de <http://www.extremeprogramming.org/>
- Yacchirema, S. A., & Ortega, I. M. (2015). *Desarrollo e Implantación de una Plataforma para controlar la comunicación en tiempo real basada en la tecnología WebRTC*. Sangolqui.
- Yoro, D. (2006). Identificación y Clasificación de Patrones en el Diseño de Aplicaciones Móviles. . *Universidad Nacional de la Plata* .
- Zeng, K. (2016). QUALITY-OF-EXPERIENCE OF STREAMING VIDEO: INTERACTIONS BETWEEN PRESENTATION QUALITY AND PLAYBACK STALLING . *IEEE* .
- Zhang, W. (2013). QoE-Driven Cache Management for HTTP Adaptive Bit Rate Streaming Over Wireless Networks. *IEEE* .

