



FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

TÍTULO: InfoMeeting: Una herramienta de comunicación sincrónica moderada

AUTORES: Pablo Martín Digiani

DIRECTOR: Esp. Gladys Mabel Gorga

CODIRECTOR: Dra. Cecilia Verónica Sanz

ASESOR PROFESIONAL:

CARRERA: Licenciatura en Sistemas

Resumen

En este trabajo, se han investigado herramientas de comunicación sincrónica orientadas a procesos educativos. Se analizaron experiencias de uso de este tipo de herramientas y tres casos de tecnologías vinculadas a este tipo de comunicación con sus características y aspectos de implementación subyacentes. Los hallazgos obtenidos motivaron la creación de InfoMeeting, herramienta que presenta características para la coordinación de reuniones educativas (como solicitar la palabra y compartir la presentación del moderador), encuestas en tiempo real, edición de código de un algoritmo, de manera colaborativa, chats y el registro de lo ocurrido en las sesiones. Se llevaron a cabo pruebas en diferentes sesiones que permitieron la mejora de la herramienta. Al mismo tiempo se trabajó en la integración de InfoMeeting con un EVEA.

Palabras Clave

Comunicación Sincrónica, Educación, Videoconferencia, Colaboración, Editor de Código, Moderación, WebRTC, WebSockets, Node.js

Conclusiones

El proceso realizado en la tesina permitió el estudio de diferentes tecnologías vinculadas a la comunicación sincrónica. El trabajo de diseño e implementación de InfoMeeting se enmarca en un desarrollo de software distribuido que da soporte a entornos heterogéneos de trabajo. InfoMeeting fue valorada positivamente en términos de usabilidad y fue considerada un complemento de interés para la comunicación entre docentes y alumnos. Su integración con un EVEA permite mantener el seguimiento de las sesiones que se realicen. Una característica que se destacó en las pruebas con usuarios fue el editor de código que resulta apropiado para cursos que trabajan con algoritmos.

Trabajos Realizados

- ❖ Un análisis de antecedentes de experiencias educativas que utilizan herramientas de comunicación sincrónica.
- ❖ Un análisis de características y tecnologías subyacentes de herramientas de comunicación sincrónica reconocidas en la comunidad educativa.
- ❖ El diseño y desarrollo de la herramienta InfoMeeting, utilizando tecnologías de WebSockets y WebRTC.
- ❖ un proceso de sesiones de prueba con el propósito de validar la herramienta en un escenario distribuido, junto con el análisis de los resultados obtenidos.
- ❖ La integración de la herramienta con el EVEA IDEAS.

Trabajos Futuros

Se proponen como trabajos futuros:

- ❖ La extensión del funcionamiento del editor para: propagar eventos de scroll, foco y selección de texto; dar soporte para el resaltado de sintaxis de múltiples lenguajes de programación y permitir elegir el lenguaje que se quiera utilizar.
- ❖ La posibilidad de permitir no sólo la escritura de código en el editor, sino también su ejecución.
- ❖ La creación de un panel de control donde se pueda monitorear el estado de la herramienta y guardar un log de errores.

InfoMeeting: una herramienta de comunicación
sincrónica moderada

Pablo Digiani

Noviembre 2018

Agradecimientos

A Cecilia y a Gladys por guiarme con su dedicación y su profesionalidad durante todo el desarrollo.

A Amanda por su apoyo incondicional durante todo el proceso.

A todo el equipo de IDEAS por su buena predisposición a la hora de integrar InfoMeeting.

A los participantes de la sesión de pruebas.

A Lautaro y a todo el equipo de Axlot por motivarme cada día, y en especial a Daiana por sus consejos para pulir la presentación.

A mis padres, que lamentablemente no pudieron ser testigos de este momento.

A Juan José, Nestor, Patricio, Matías, Esteban, Gabriel, Diego, Adrián, Alejandra, Roberto y otros tantos compañeros más. Gracias a ellos por hacer que la carrera haya sido una experiencia maravillosa e inolvidable.

Índice general

1. Introducción. Motivación y propuesta	1
1.1. Motivación	1
1.2. Contribuciones	3
1.3. Organización de la Tesis	4
2. La comunicación sincrónica en la educación	5
2.1. Introducción	5
2.2. Antecedentes	5
2.2.1. Aprendiendo un segundo idioma	6
2.2.2. Clases sincrónicas virtuales en la Patagonia	10
2.2.3. Adopción de nuevas tecnologías de enseñanza en la Universidad	14
2.3. Conclusión	18
3. Herramientas de comunicación sincrónica	20
3.1. Introducción	20
3.1.1. Selección de herramientas y aspectos a comparar	21
3.2. Apache Open Meetings	22
3.2.1. Acerca de la plataforma	22

3.2.2. Modos de interacción	24
3.2.3. Administración y sesiones	27
3.2.4. Respecto de la Experiencia de Usuario	29
3.2.5. Integración con otros sistemas	29
3.2.6. Otras características	30
3.3. BigBlueButton	31
3.3.1. Acerca de la plataforma	31
3.3.2. Modos de interacción	33
3.3.3. Administración y sesiones	35
3.3.4. Respecto de la Experiencia de Usuario	37
3.3.5. Integración con otros sistemas	38
3.3.6. Otras características	39
3.4. WebEx	39
3.4.1. Acerca de la plataforma	39
3.4.2. Modos de interacción	40
3.4.3. Administración y sesiones	42
3.4.4. Respecto de la Experiencia de Usuario	43
3.4.5. Integración con otros sistemas	44
3.4.6. Otras características	46
3.5. Conclusiones	46
4. InfoMeeting: descripción de su diseño	48
4.1. Introducción	48
4.2. Motivacion	49

4.3. Descripción general de InfoMeeting	50
4.3.1. Roles de usuarios	50
4.3.2. Componentes de la interfaz	51
4.3.3. Interfaz para dispositivos móviles	59
4.3.4. Funcionalidades principales implementadas	60
4.3.5. Persistencia de datos	65
4.3.6. Funcionalidades que no se implementaron	65
4.4. Conclusión	65
5. Aspectos Tecnológicos de Infomeeting	66
5.1. Introducción	66
5.2. Análisis de las herramientas y tecnologías subyacentes para la implementación de InfoMeeting	67
5.3. La arquitectura de Infomeeting	68
5.3.1. Capa de lógica	68
5.3.2. Capa de persistencia	75
5.3.3. Capa de presentación	77
5.4. La estructura de archivos de InfoMeeting	81
5.5. Integración con IDEAS	84
5.6. Persistencia de datos	85
5.7. API para obtener datos	86
5.8. Conclusión	88
6. Sesiones de prueba de InfoMeeting	89
6.1. Introducción	89

6.2. Etapa de Sesiones Informales	90
6.3. Sesión de prueba formal	95
6.3.1. Introducción	95
6.3.2. Reglas de participación y dinámica de la sesión	96
6.3.3. Problemas encontrados durante la sesión	97
6.3.4. Encuesta	97
6.3.5. Parte 4 - Acerca de la performance	103
6.3.6. Anexo para el Tutor	106
6.4. Conclusión	107
7. Conclusión y trabajos futuros	109
7.1. Introducción	109
7.2. Aportes realizados y Conclusiones	109
7.3. Líneas de trabajo futuro	111
Anexos	112
A. WebRTC	113
A.1. Arquitectura	114
A.2. Session description protocol	115
A.3. Network Address Translation	116
A.4. Atravesando barreras	117
A.4.1. STUN	117
A.4.2. TURN	118
A.5. ICE	119

A.6. SCTP	120
B. Encuesta a los participantes de la sesión de pruebas	121
B.1. Acerca de Usted	121
B.2. Acerca de Infomeeting	121
B.3. Acerca de la interfaz y la experiencia de usuario	122
B.4. Acerca de la performance	122
B.5. Preguntas sólo para el tutor	123
C. Guiones utilizados durante las pruebas	124
C.1. Pasos para el tutor	124
C.1.1. Conexión: pruebas de audio y video	124
C.1.2. Dar la palabra	124
C.1.3. Consultas y prueba de chat individual	125
C.1.4. Modo editor de código	125
C.1.5. Uso libre de la herramienta, prueba de bloquear participación	125
C.2. Pasos para el resto de los participantes	126
C.2.1. Conexión: pruebas de audio y video	126
C.2.2. Dar la palabra	126
C.2.3. Consultas y prueba de chat individual	126
C.2.4. Modo editor de código	126
C.2.5. Uso libre de la herramienta	126

Índice de figuras

2.1. Profesora enseña a sus alumnos a trazar caracteres usando 3C.	8
2.2. Infraestructura de la solución propuesta.	13
2.3. El aula virtual de Horizon Wimba.	16
3.1. Pizarra y panel de usuarios en AOM. Los moderadores pueden otorgar permisos a los demas usuarios para compartir pantalla, usar el whiteboard, camara y micrófono.	26
3.2. Sesiones grabadas en una instalacion de AOM	27
3.3. El calendario de AOM muestra los eventos agendados	28
3.4. Al momento de entrar a una sesión podemos elegir entre dos modos de participación	33
3.5. BBB permite enviar mensajes de chat privados	34
3.6. Pizarra o <i>Whiteboard</i> de BBB, compartiendo un archivo .doc	35
3.7. Pantalla principal de BigBlueButton: a la izquierda se observa el panel de usuarios con tres participantes conectados, en el centro se muestra un documento ppt compartido y a la derecha se encuentra el panel de chat.	37
3.8. Interfaz de usuario de WebEx: a la izquierda se observa la pizarra colaborativa y a la derecha el panel de gestión de usuarios.	41
3.9. Barra de menú WebEx en modo pantalla completa.	44
3.10Arquitectura de la API XML de WebEx. Los servicios XML proveen funciones muy útiles como gestionar usuarios o programar reuniones.	45

4.1. Diagrama de casos de uso de InfoMeeting. Se observan dos roles de usuario diferentes.	51
4.2. Interfaz de usuario de Infomeeting. La interfaz compone de varios paneles. El componente principal (izquierda) se encuentra en modo editor de código.	52
4.3. El componente de video principal	53
4.4. Esta pantalla es un ejemplo de cómo el tutor de la sesión utiliza el editor de código para explicar un algoritmo	54
4.5. El panel de usuario visto por un docente moderador (izquierda) y por un alumno (derecha)	55
4.6. El moderador recibe notificaciones de un nuevo mensaje proveniente del usuario Maria Correa	56
4.7. Panel de chat general	57
4.8. Chat privado entre dos alumnos	58
4.9. Componente de video secundario. Sólo se muestra cuando el usuario no tiene la palabra o cuando la herramienta está en modo editor de texto.	58
4.10InfoMeeting vista desde un dispositivo móvil.	59
4.11Menú de la aplicación en versión móvil.	60
4.12El usuario puede pedir el control (o devolverlo en caso de que ya lo tuviera) haciendo clic en el icono del megáfono en el menú superior.	61
4.13El usuario recibe una confirmación de que su pedido de palabra ha sido enviado.	61
4.14El moderador (Pablo) observa que los alumnos Jorge y Maria le piden la palabra.	62
4.15El moderador recibe una confirmación de que ha transferido el control de la sesión.	62
4.16El moderador de la sesión crea una encuesta para ser respondida por los alumnos.	63
4.17El alumno recibe una notificación de que posee encuestas sin responder.	63
4.18Ventana modal de una encuesta. El alumno puede responder por verdadero o falso.	64
4.19Confirmación de respuesta a la encuesta.	64

5.1. Arquitectura de la plataforma InfoMeeting.	68
5.2. Ciclo de eventos utilizado por Node.js.	69
5.3. Definición de rutas en Express.js	70
5.4. Definición de un Middleware en Express.js	71
5.5. Representación visual de una conexión mediante WebSockets.	72
5.6. Mensajes recibidos por el cliente de InfoMeeting.	73
5.7. Mensajes recibidos por el servidor de InfoMeeting.	74
5.8. Conexión a la base de datos con la librería mysql.	76
5.9. Comparación de una consulta hecha con <i>callbacks</i> (parte superior) y con <i>promises</i> (parte inferior).	77
5.10 Algunos de los componentes que ofrece del <i>framework</i> Bootstrap.	80
5.11 El archivo main.js inicializa los eventos de la interfaz.	82
5.12 Estructura de la clase InfoMeeting	83
5.13 El archivo config.js permite definir los parámetros básicos de InfoMeeting.	84
5.14 Integración entre IDEAS e InfoMeeting.	84
5.15 Token recibido por InfoMeeting	85
5.16 Modelo de la base de datos de InfoMeeting.	86
5.17 Información expuesta por InfoMeeting.	87
6.1. Así se veía el panel de participantes antes de la 5ta sesión de prueba. Finalmente, se decidió reemplazar los avatares estáticos por un componente de video.	95
6.2. Uno de los participantes durante la sesión de prueba.	96
6.3. Respuestas a la pregunta <i>Indique cuales de las siguientes herramientas de comu- nicación de propósito general ha utilizado.</i>	98

6.4. Respuestas a la pregunta <i>Indique si conoces alguna de estas herramientas que fueron utilizadas como inspiración para crear InfoMeeting.</i>	98
6.5. Respuestas a la pregunta <i>¿Qué funciones le parecieron mas útiles?</i>	99
6.6. Respuestas a la pregunta <i>¿Qué tan útil le pareció el editor de código?</i>	99
6.7. Respuestas a la pregunta <i>¿Considera que InfoMeeting puede ser un complemento apropiado para la comunicación entre alumnos y docentes durante el dictado de un curso?</i>	100
6.8. Respuestas a la pregunta <i>¿Qué tan intuitiva le resultó la interfaz (estructura, organización, etc)?</i> . <i>El rango de respuestas iba desde poco intuitiva hasta muy intuitiva.</i>	101
6.9. Respuestas a la pregunta <i>¿Considera que las fuentes utilizadas, así como el tamaño de las mismas es adecuada?</i> . <i>El rango de respuestas iba desde poco adecuadas hasta muy adecuadas.</i>	101
6.10 Respuestas a la pregunta <i>¿Considera que los colores utilizados tienen un buen contraste y favorecen la legibilidad?</i> . <i>El rango de respuestas iba desde muy en desacuerdo hasta muy de acuerdo.</i>	102
6.11 Respuestas a la pregunta <i>¿Qué tan intuitivos le resultaron los íconos?</i> . <i>El rango de respuestas iba desde poco intuitivos hasta muy intuitivos.</i>	102
6.12 Respuestas a la pregunta <i>¿Considera que las notificaciones sonoras al recibir mensajes son apropiadas?</i> . <i>El rango de respuestas iba desde muy en desacuerdo hasta muy de acuerdo.</i>	103
6.13 Respuestas a la pregunta <i>¿Cómo le resultó el tiempo de carga y de respuesta la aplicación?</i> . <i>El rango de respuestas iba desde muy lento hasta muy rápido.</i>	104
6.14 Respuestas a la pregunta <i>¿Cómo notó la calidad del video?</i> . <i>El rango de respuestas iba desde muy baja hasta muy alta.</i>	104
6.15 Respuestas a la pregunta <i>¿Cómo notó la calidad del audio?</i> . <i>El rango de respuestas iba desde muy baja hasta muy alta.</i>	105
6.16 Respuestas a la pregunta <i>¿Qué dispositivo utilizó?</i>	105
6.17 Respuestas a la pregunta <i>¿Qué tipo de conexión a internet utilizó?</i>	106

6.18 Preguntas y respuestas para el tutor. Las escalas utilizadas iban desde <i>Muy difícil</i> a <i>Muy fácil</i> en la primer y tercera pregunta, mientras que para la segunda se uso un rango desde <i>Poco útiles o escasas</i> hasta <i>Muy útiles</i>	107
A.1. Arquitectura de la tecnología WebRTC	114
A.2. Un ejemplo de mensaje SDP. El parámetro c describe los datos de la conexión. . . .	116
A.3. NAT permite comunicar varios dispositivos locales con direcciones ip externas . . .	117
A.4. Comunicación utilizando un Servidor STUN	118
A.5. Comunicación utilizando un Servidor TURN	119
A.6. Pila de protocolos en una aplicacion WebRTC	120

Índice de tablas

6.2. Respuestas a la pregunta <i>¿Qué fue lo que más les gustó de InfoMeeting?</i>	99
6.3. Respuestas a la pregunta <i>¿Qué fue lo que menos le gustó InfoMeeting?</i>	100
6.4. Respuestas a la pregunta <i>¿Qué funcionalidad le agregaría a la herramienta?</i>	100
6.5. Respuestas a la pregunta <i>¿Qué aspectos de la interfaz cree que pueden mejorarse?</i>	103
6.6. Respuestas a la pregunta <i>¿Qué Sistema Operativo y navegador utilizó?</i>	105

Capítulo 1

Introducción. Motivación y propuesta

1.1. Motivación

Los términos sincrónicos y asincrónicos aparecen una y otra vez en el contexto del aprendizaje a distancia. Collins y Berge (1994) definieron la comunicación sincrónica como la actividad comunicativa que ocurre en tiempo real como en el caso de la comunicación presencial [...], y los participantes en interacción deben estar presentes, aunque no necesariamente en la misma localización física. Por otro lado, la comunicación asincrónica [...] se encuentra mediada tecnológicamente y no depende de que los estudiantes y profesores estén presentes al mismo tiempo para dirigir las actividades de enseñanza y aprendizaje.

Branon y Essex (2001) señalan algunas desventajas de la comunicación asincrónica como la falta de retroacción inmediata a los mensajes individuales, la necesidad de participación diaria de los alumnos, así como la cantidad de tiempo necesario para el desarrollo de un debate de calidad, estos aspectos pueden contribuir a que la participación se resienta. Así, muchos alumnos pueden desvincularse porque sus ideas expuestas, por ejemplo en un foro, no reciben respuesta. Por otra parte, las discusiones en línea se extienden a lo largo de más tiempo que las conversaciones cara a cara. Por ello, algunos participantes expresan dificultades al tener que manejar el volumen de mensajes que se producen en estos debates. Sin embargo, la comunicación asincrónica tiene otras ventajas que son aprovechadas hoy día en numerosas propuestas educativas, y se trata de complementar con instancias de comunicación sincrónica para salvar

los aspectos que pueden ser negativos.

Por su parte, la comunicación sincrónica tampoco es apropiada para todas las situaciones, pero aporta otras alternativas de diálogo entre docentes y alumnos y alumnos entre sí, en modalidades educativas mediadas por tecnologías digitales. Una de las principales ventajas de este tipo de comunicación se vincula con la posibilidad de obtener un *feedback* inmediato de los participantes, a partir de la interacción en tiempo real. Aunque, se dificulta la coordinación de un tiempo coincidente para la participación sincrónica. Otras características que resultan beneficiosas de las herramientas de comunicación sincrónicas son: la facilidad para realizar encuestas en tiempo real, la posibilidad de que las reuniones sean coordinadas por un moderador simulando el aula física de clase, el registro de lo ocurrido en la reunión, la planificación de los encuentros con la disponibilidad del material (presentaciones, videos, etc.), entre otras.

Llorente y Cabrero (2007) afirman que la posibilidad de revisar el registro histórico de la comunicación realizada facilita la comunicación entre personas de diferente espacio geográfico, y favorece la transferencia de conocimiento. Además, frente a la comunicación verbal, la escrita brinda la posibilidad de revisar el discurso por todos los participantes, y a partir de ahí, establecer conclusiones sobre varios aspectos, por ejemplo: cómo se ha generado el discurso, cuáles han sido las aportaciones de los diferentes miembros, cómo se ha formado el conocimiento experto, o cuáles han sido las reglas que han dirigido la interacción.

Valverde-Berrocoso (2002) menciona que las herramientas de comunicación sincrónicas más utilizadas son las siguientes:

- IRC, Internet Relay Chat o simplemente chat. Se trata de un medio de comunicación que permite realizar conversaciones en tiempo real a través de internet. Aunque normalmente se utiliza para conversar en línea también sirve para transmitir ficheros. El IRC fue diseñado originalmente por Jarkko Oikarinen en el verano de 1988. Los primeros servidores se establecieron en Finlandia y, desde allí, se propagaron por más de 60 países.
- Videoconferencia. Una videoconferencia conecta dos o más sitios con enlaces de audio y video de forma que los participantes pueden conversar, interactuar y compartir documentos. Como herramienta de educación a distancia, las videoconferencias ayudan a los docentes a comunicarse con estudiantes aislados por razones espaciales y/o temporales; permiten conectar aulas para la realización de proyectos de colaboración o comunicar a profesores en formación con expertos en contenidos y metodologías didácticas.
- Los MUD (Multi-User Dimensions, Multi-User Domains o Multi-User Dungeons) se los define como un mundo virtual, construido fundamentalmente a través de textos escritos que

se elaboran en línea a modo de chat, en el que los usuarios adoptan una personalidad definida por unas determinadas características (*roleplaying*), e interactúan en dicho espacio según los principios y normas que lo rigen.

Respecto de los chats, si bien existen múltiples implementaciones usando diferentes tecnologías y plataformas, la mayoría carece de herramientas para facilitar la moderación de las sesiones. Surge la idea, entonces, de facilitar la conducción de las sesiones a los usuarios moderadores, para que estos tengan la capacidad de determinar quién tendrá la palabra en un determinado momento, y que de esa manera la comunicación se realice de forma organizada y no se convierta en una mala experiencia para los participantes. Si bien existen algunas herramientas disponibles con estas facilidades, no se integran fácilmente con los Entornos Virtuales de Enseñanza y Aprendizaje (EVEA) actuales.

Al integrar la herramienta con un EVEA, quien organiza las reuniones sincrónicas tiene la posibilidad de preparar tópicos de conversación específicos que se lleven a cabo en un día determinado, para que los alumnos puedan visualizar los próximos encuentros con anticipación. Esto ayuda a la planificación de docentes y alumnos, dándole más seriedad y tiempo de preparación a los participantes y moderadores. A su vez, esta integración favorece el seguimiento y trazabilidad de las actividades de los docentes y de los alumnos en el espacio del curso.

Es por ello que para este trabajo, se propone el desarrollo de una herramienta de comunicación sincrónica (InfoMeeting) que presente características para la coordinación de las reuniones (solicitar la palabra, compartir la presentación del moderador. . .), la realización de encuestas en tiempo real, el *feedback* inmediato y el registro de lo ocurrido en las sesiones. Al mismo tiempo, se propone generar una estrategia de integración con un EVEA específico de manera tal de favorecer la trazabilidad de las actividades de los participantes.

1.2. Contribuciones

Con este trabajo se pretende:

- Brindar un análisis de herramientas de comunicación sincrónica existentes.
- Ofrecer la herramienta InfoMeeting integrada en un EVEA.
- Mostrar un análisis de usabilidad de la herramienta desarrollada.

Al mismo tiempo se espera que, a partir de este trabajo, la herramienta quede como un producto provechoso tanto para los alumnos como para los docentes que la utilicen.

1.3. Organización de la Tesis

El trabajo está dividido en 7 capítulos principales:

- Capítulo 2: La comunicación sincrónica en la educación. Se estudian las diferentes herramientas encontradas y se realiza una comparación de sus características, focalizándose en diversos aspectos generales como el tipo de licencia, las plataformas en las que se encuentran disponibles y también en aspectos funcionales como la posibilidad de tener chats grupales, de contar con moderadores, o de realizar encuestas entre otros.
- Capítulo 3: Herramientas de comunicación sincrónica. Se brinda un análisis técnico de las tecnologías y protocolos utilizados para implementar las herramientas descriptas en el capítulo anterior.
- Capítulo 4: InfoMeeting: descripción de su diseño. Se presentan las características InfoMeeting, sus requerimientos funcionales y casos de uso tanto para usuarios participantes como para usuarios moderadores.
- Capítulo 5: Aspectos tecnológicos de InfoMeeting. Se describe la implementación de InfoMeeting, las dificultades encontradas durante el desarrollo, las tecnologías utilizadas y la arquitectura de la plataforma.
- Capítulo 6: Sesiones de prueba con InfoMeeting. Se analiza la usabilidad de la herramienta describiendo las experiencias realizadas entre alumnos y profesores. Se determinan ciertos parámetros a medir y exponen las conclusiones obtenidas.
- Capítulo 7: Trabajo futuro y conclusiones. Se proponen distintas extensiones para enriquecer el uso de la herramienta.

Capítulo 2

La comunicación sincrónica en la educación

2.1. Introducción

En este capítulo se presentan algunos antecedentes de experiencias educativas usando herramientas de comunicación sincrónica. En cada experiencia se describe el dominio de la misma, así como las tecnologías utilizadas para realizar la experiencia, la metodología llevada a cabo y los resultados obtenidos.

2.2. Antecedentes

En esta sección se describirán tres experiencias en las que la comunicación sincrónica adquiere relevancia.

La primera de ellas, consiste en la utilización de una herramienta de comunicación sincrónica para enseñar un segundo idioma. La segunda y la tercera detallan pruebas realizadas utilizando diferentes herramientas de comunicación sincrónica en universidades, una de ellas en Argentina y otra en Estados Unidos.

En cada experiencia se detalla el marco contextual en el que fue realizada, así como los motivos que impulsaron el uso de tales prácticas. Además se describe herramienta utilizada

y la metodología empleada en la evaluación de la misma. Las experiencias concluyen con una sección de resultados donde se resumen los hallazgos obtenidos durante el proceso.

Este análisis de experiencias permitirá al mismo tiempo reconocer qué tipo de prácticas se llevan adelante con herramientas de comunicación sincrónicas. Asimismo se observará si existen algunas barreras en el uso de estas herramientas.

2.2.1. Aprendiendo un segundo idioma

Introducción

Aprender un segundo idioma a distancia supone un gran desafío para los estudiantes. La distancia física de los profesores y compañeros, junto con el posible aislamiento de la comunidad lingüística pertinente, dificulta enormemente el aprendizaje de idiomas. Sin embargo, la interacción en línea apoyada por sistemas de gestión de aprendizaje sincrónico o *Second Language Management Systems* (SLMS) a través de internet tiene un gran potencial para abordar varias dificultades que enfrentan los estudiantes de idiomas a distancia. Al permitir que estudiantes e instructores aborden juntos los problemas lingüísticos en entornos virtuales en tiempo real, las funciones interactivas de estos sistemas (chats, pizarras y videoconferencia) pueden ayudar a fomentar comunidades de aprendizaje vitales en la enseñanza de un segundo idioma.

El caso presentado ofrece una ilustración para apoyar la enseñanza a través de entornos virtuales de aprendizaje, destacando la importancia de la interacción sincrónica

Motivación

A pesar de contar con muchas ventajas, incluyendo la flexibilidad y los bajos costos, la educación a distancia tiene a menudo desventajas, como la falta de comunicación, el sentimiento de aislamiento y la falta de confianza comúnmente sentida por los estudiantes. Los aprendices de idiomas a distancia tienen pocas oportunidades de comunicarse en el propio idioma dentro de este ambiente particular de aprendizaje; estos mismos estudiantes se sienten muy frustrados cuando no pueden conversar espontáneamente en situaciones cara a cara. En respuesta, los educadores de idiomas a distancia y los investigadores han intentado encontrar una solución a este problema, experimentando con diversas tecnologías como cintas de audio y video, teléfono, paquetes multimedia y herramientas de software en línea (como correo electrónico y videoconferencias). Estas tecnologías pueden favorecer y acompañar las necesidades de los

estudiantes en estos tipos de procesos educativos mediados.

Herramienta utilizada

La herramienta utilizada en la experiencia, fue una plataforma desarrollada por la Universidad Nacional Yat-sen (Taiwan) llamada Cyber Comunidad Colaborativa (*Collaborative Cyber Community*), también conocida como 3C¹. Esta plataforma fue desarrollada inicialmente en 1997 por el Consejo Nacional de Ciencia de Taiwan y, desde ese entonces, mantenida por un grupo de expertos de Yat-sen, liderado por el profesor Nian-Shing Chen. En términos de escalabilidad, el servidor que contenía la plataforma, poseía una capacidad de hasta 200 usuarios conectados.

La plataforma 3C se compone de dos ambientes principales. Uno de ellos, llamado oficina del profesor, sólo puede ser accedido por el profesor y puede ser utilizado para planear actividades, subir recursos de aprendizaje y enlaces. El otro, el salón de clases, es accesible tanto por el profesor como por los alumnos, e incluye un modo sincrónico y otro asincrónico.

El modo asincrónico presenta un espacio de aprendizaje para los participantes, donde por un lado, pueden acceder a diferentes recursos como notas, material del curso, ejercicios y grabaciones, y por otro lado participar en comunicaciones escritas mediante foros de discusión y mensajería.

El modo sincrónico cuenta con interacción de audio y video, herramientas para compartir información, un panel de control y múltiples salones virtuales.

Para lograr la interacción con audio/video, la plataforma utiliza una herramienta de escritorio llamada JointNet. Esta herramienta le permite al profesor ver y hablar con cuantos alumnos desee simultáneamente, logrando así, crear dentro de la plataforma, una dinámica similar a la que se ve en un aula tradicional. Los estudiantes pueden hablarle al profesor y al resto de la clase presionando el botón *talk*. Además cuenta con una caja de texto debajo del video, donde los participantes pueden chatear a medida que se desarrolla la clase.

Como complemento, el aula virtual cuenta con varias herramientas para compartir información, convirtiendo al salón en un ambiente multimodal y más completo que un aula tradicional.

Más allá de las funciones de videoconferencias, la herramienta más valiosa es la pizarra o *whiteboard* digital (Figura 2.1), que funciona de una manera similar a un pizarrón tradicional, donde se puede escribir libremente pero además permite cargar y mostrar documentos durante

¹<http://ccc.k12.edu.tw/>

el transcurso de la clase. Al aprender un segundo idioma, la pizarra cumple un rol importante a la hora de aprender a trazar caracteres de otros alfabetos, y le permite a los alumnos y profesores hacerlo simultáneamente.

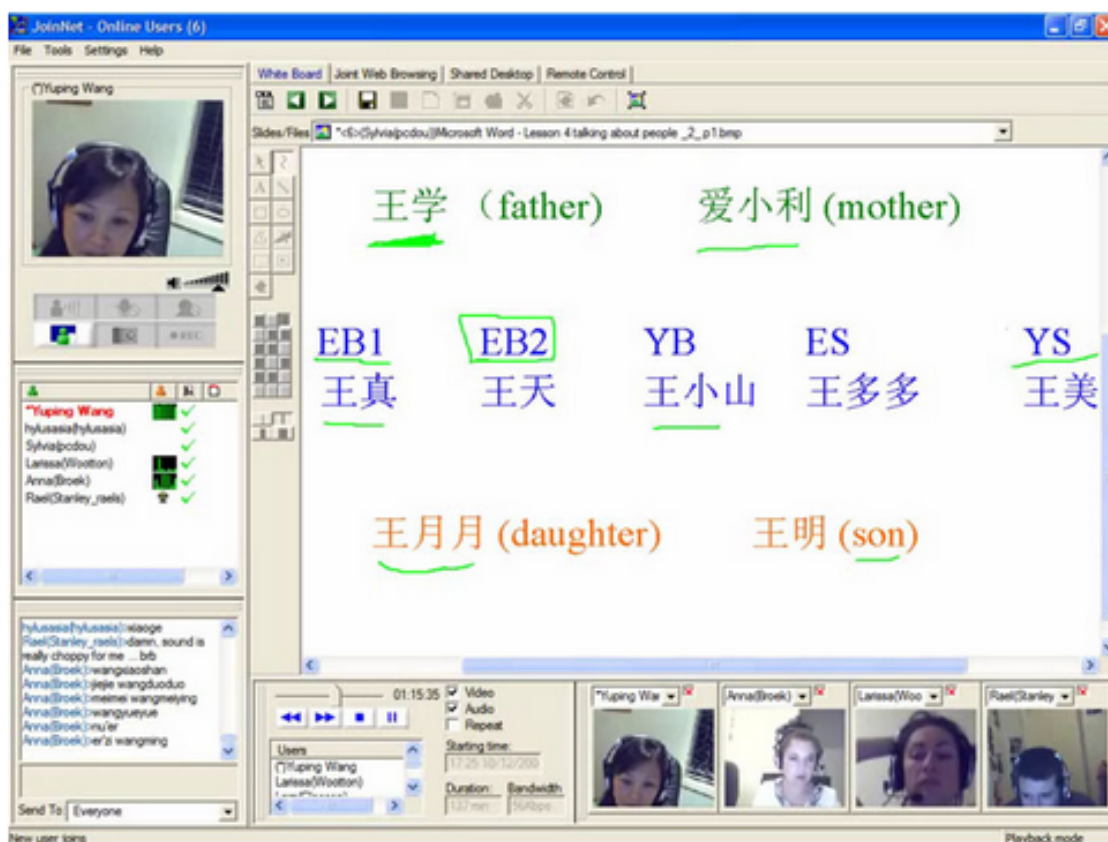


Figura 2.1: Profesora enseña a sus alumnos a trazar caracteres usando 3C.

Existe un panel de control en la plataforma 3C, donde se muestran los nombres de los estudiantes a medida que estos ingresan al salón. Este panel brinda a los profesores un acceso rápido a varias acciones sobre las personas seleccionadas, incluyendo preguntas individuales a los estudiantes, controles adicionales (por ejemplo, para compartir el escritorio), participación en encuestas y la posibilidad de desconectar a los participantes.

Dado que cuenta con varias salas simultáneas, los estudiantes pueden organizarse en diferentes grupos y practicar algún tema de interés. El profesor puede navegar por estas salas para observar y guiar las discusiones. Esta característica permite al profesor estar atento a las diferentes necesidades del alumno y también facilita la práctica en grupos pequeños.

Todas las actividades sincrónicas pueden ser grabadas utilizando una herramienta de grabación incorporada en la plataforma 3C. La grabación de vídeo se puede publicar en el sistema,

para que los alumnos puedan reproducirlos después de la clase para revisarlos o para ponerse al día con las clases que podrían haber perdido.

Metodología

Un estudio piloto se llevó a cabo a finales de 2005 para examinar la eficacia de la plataforma 3C. Siete estudiantes intermedios del Programa Abierto de Aprendizaje de Chino en la Universidad de Griffith (Australia) participaron en el estudio. Cinco de ellos eran de Australia, uno era de Hong Kong y otro de República Checa. Cada participante recibió, además del acceso al sistema, una cámara Web y un auricular.

A través de la plataforma 3C, se llevó a cabo una clase en vivo de dos horas cada semana durante 10 semanas. Se requirió a los participantes que completaran las tareas de aprendizaje, ya sea en forma de juego de roles, juego o diálogo en el aula sincrónica. Los participantes usualmente realizaban la tarea oralmente uno con otro, y el chat de texto se utilizó en chino e inglés para complementar las actividades. Todas las actividades fueron grabadas y se colocaron en el sistema para la revisión de los participantes.

Después de las clases, los participantes fueron encuestados por correo electrónico. Dado el número reducido de participantes, las preguntas de la encuesta fueron de composición abierta en lugar de múltiples opciones.

Resultados

Los resultados de la encuesta demostraron que todos los participantes adoptaron con satisfacción y entusiasmo el aprendizaje sincrónico y afirmaron que disfrutaron de la experiencia.

Algunos participantes notaron progreso en su aprendizaje como resultado y uno de ellos consideró que "mejoró dramáticamente en un plazo muy corto".

Esta experiencia positiva, también llevó a una mayor confianza en la adquisición del idioma y un sentido de conexión entre los participantes.

La conveniencia y la eficiencia en el tiempo, fueron otras dos ventajas importantes citadas por los participantes. Esto es probablemente porque todos los participantes asistieron a las clases desde sus hogares, ahorrando así tiempo para viajar. Un participante también opinó que esta forma de aprendizaje podría crear menos ansiedad que cara a cara ya que "hay un cierto grado de anonimato".

Un participante consideró muy buenas las grabaciones de las clases, observando que estas grabaciones podían ser reproducidas para analizar los errores o entender cosas que uno se perdió durante las sesiones. Esta observación apunta a la eficacia de un modelo de aprendizaje que combina el aprendizaje sincrónico con recursos asíncronos.

Algunos participantes se frustraron por la calidad del sonido y problemas técnicos. La calidad de sonido de las clases síncronas era similar a la de una llamada telefónica internacional si se utilizaba la banda ancha. Sin embargo, la calidad se reducía notablemente cuando se utilizaba una conexión de acceso telefónico.

A pesar de estos problemas, todos los participantes sugirieron la inclusión de este tipo de aprendizaje en sus programas a distancia en el futuro y, en general, su entusiasmo fue evidente.

2.2.2. Clases sincrónicas virtuales en la Patagonia

Introducción y motivación

Este antecedente histórico, cuenta la experiencia desarrollada en el dictado del curso Sistemas Distribuidos entre la Universidad Nacional del Comahue (U.N.Co.) y la Universidad Nacional de la Patagonia Austral (U.N.P.A.) en el año 2007. Ambas universidades patagónicas comparten características como la dispersión geográfica de sus asentamientos, la distancia de los grandes centros urbanos donde se da la mayor oferta de cursos de actualización y perfeccionamiento y recursos humanos en formación, por lo que era necesario optimizar la cantidad y calidad de capacitación. En cuanto a la experiencia en el uso de plataformas de educación a distancia, en la mayoría de las cátedras sólo se la había utilizado como repositorio de material, aunque existía una fuerte política universitaria en ambas casas de altos estudios en fomentar la bimodalidad.

En este contexto y teniendo en cuenta los objetivos planteados se diagramó una propuesta de dictado a distancia del curso Sistemas Distribuidos. La edad del estudiantado al que fue dirigido oscilaba entre los 20 y 35 años de edad, sin experiencia previa en esta metodología, cuya motivación para acreditar el curso estaba basada tanto en la preparación básica profesional, como en la actualización de conocimientos.

Metodología y herramientas utilizadas

Dicha propuesta estableció dos visitas de los profesores. La primera al inicio de la cursada con el objeto de realizar la presentación de la cátedra, que los integrantes puedan conocerse y que éstos se sientan familiarizados con la forma de trabajo. En este primer encuentro no se tenía como tarea el dictado de clases presenciales, sino que se realizaron trabajos de integración grupal y aproximación a las tareas. Luego se ultimaron detalles de la configuración del equipamiento que formaría la infraestructura sobre la que se montaría la experiencia. Finalmente, durante este primer encuentro, se realizó una experiencia piloto de una clase sincrónica a distancia. La segunda visita que se realizó al final del cursado, tuvo como objetivo evaluar a los alumnos.

El dictado del curso estuvo soportado por TICs a través de la plataforma de educación a distancia PEDCO, implementada sobre el sistema de manejo de cursos Moodle. PEDCO es una herramienta que dispone de variados mecanismos de comunicación y que permite el aprendizaje cooperativo y colaborativo. Esta herramienta era conocida por la mayoría de los participantes, lo que facilitó su uso. La página del curso se organizó siguiendo un diagrama por temas, en donde se publicaron las transparencias de las teorías, los prácticos a resolver, enlaces a los sitios desde donde se puede bajar el software, de carácter *freeware*, requerido para resolver los prácticos, enlaces a tutoriales y material de consulta correspondientes a la temática abordada.

De los recursos de comunicación asincrónica de la plataforma el más utilizado fue el foro. Esta herramienta permitió mejorar la comunicación, manteniendo un vínculo permanente docente-alumno y alumno-alumno, como así también permitió crear debates entre los diferentes actores.

En el término de los meses que duró el curso se planificaron dos encuentros semanales virtuales sincrónicos. Los docentes dictaron durante uno de esos encuentros, clases teórica de dos horas de duración. En cada una de ellas, el docente desarrolló un tema a través de diapositivas. Los alumnos y docentes en formación siguieron la clase a través del avance automático de la presentación y de la explicación del profesor. Los estudiantes pudieron resolver sus dudas en el momento de la presentación. En esta primera experiencia los alumnos tuvieron disponible en la plataforma las diapositivas aunque no se mantuvo un registro del audio de las clases. Sin embargo, los alumnos podían recuperar las comunicaciones realizadas por Chat.

Las prácticas fueron diagramadas como de laboratorio y se planificaron a través de actividades de naturaleza obligatoria. El docente una vez introducido el tema presentaba un trabajo que los alumnos resolvieron y defendieron. La evaluación constó de la defensa de los trabajos

realizados, cuya presentación sin errores conceptuales graves habilitó al alumno a rendir un examen oral.

Con respecto a los aspectos técnicos subyacentes a la comunicación sincrónica, se planificó el uso de herramientas para el intercambio de imágenes como NetMeeting o Ekiga, pero el ancho de banda disponible entre las instituciones no alcanzaba a satisfacer parámetros de calidad aceptable en la señal, por lo que se optó por dar solamente soporte de audio. Así, en esta primer experiencia se utilizó Skype, un software gratuito que tiene uso similar a un teléfono por internet y que presta servicio de mensajería. Skype se encuentra disponible para cualquier plataforma. Otro aspecto a solucionar fue compartir los escritorios, ya sea para que el docente dicte su clase o bien para que el alumno pudiera mostrar la implementación de su trabajo práctico y hacer las consultas directamente sobre su resolución.

La modalidad elegida fue la de WebCast, cuya característica principal es su facilidad de uso, tanto en la generación del material, como en su despliegue sobre los servidores Web. WebCast se utilizó para la difusión del material didáctico en las clases teóricas. El docente preparó su clase teórica con diapositivas y las exportó con un formato WebCast. Esto le permite al docente controlar la secuencia de diapositivas que se muestra, de modo que cada vez que cambia de diapositiva se modifica, en forma transparente y simultánea, la presentación a cada uno de los alumnos.

El servidor que alojaba las presentaciones se encontraba en las instalaciones de la U.N.P.A., por una cuestión de proximidad geográfica y mejora en el tiempo de respuesta. El docente controlaba la presentación desde la U.N.Co. en Neuquén y los alumnos en sus casas y en instalaciones de la U.N.P.A. observaban la presentación, escuchaban lo que el docente exponía y participaban activamente de la clase a través de una conferencia sobre Skype.

Para compartir el escritorio de trabajo, y poder acceder y participar en forma interactiva de las aplicaciones que se estén ejecutando en la máquina remota, el software seleccionado fue VNC. Este software permite el control remoto de una computadora, de modo que el docente pudo asistir al alumno en las clases prácticas, diagramadas con tareas de programación. Una vez que el docente comenzaba sus tareas, podía acceder al equipo del alumno y observar el proceso de resolución, así como también acceder directamente sobre el código que el alumno se encontraba escribiendo.

En la Figura 2.2 se puede observar como se dispuso la infraestructura de la solución. En el recuadro U.N.Co. se representa al docente y a los servicios ofrecidos: plataforma de educación a distancia y servidor Web. Los restantes recuadros representan a otros asentamientos de la

U.N.P.A. y a alumnos imposibilitados de asistir a la U.A.R.G. que siguieron la clase desde lugares físicos diferentes. A excepción de las conexiones prestadas por la U.N.Co. y la U.N.P.A., que disponían de ancho de banda simétrico, el resto de las conexiones utilizaron ADSL.

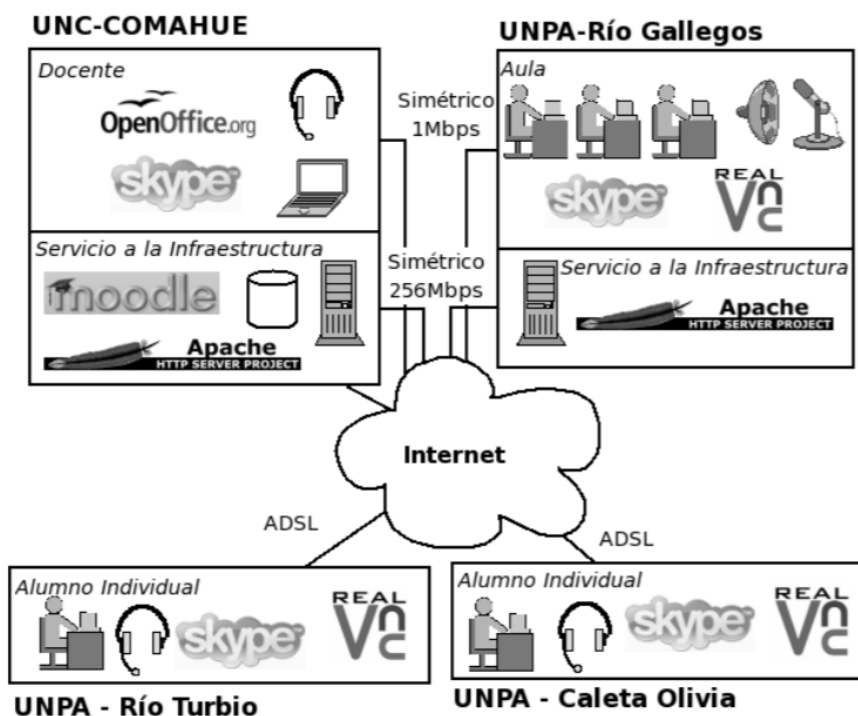


Figura 2.2: Infraestructura de la solución propuesta.

Resultados

La experiencia presentó algunas falencias propias de los problemas de infraestructura. Entre ellas el ancho de banda en la conexión el cual no permitió incorporar video a la clase virtual, por no contar con los requerimientos tecnológicos mínimos necesarios. Visualizar al profesor es importante al momento de extender una explicación sobre un tema complejo, ya que permite el uso de otras habilidades y herramientas que son de uso común en una clase presencial. Los alumnos que se saben visualizados por el profesor aumentan su concentración en la clase misma. Además, al igual que en una clase tradicional presencial, disminuye el anonimato del estudiante, pues el profesor puede saber exactamente en cada momento con que audiencia cuenta y que nivel de entendimiento logran.

Los alumnos siguieron sin problemas el desarrollo de las clases teóricas a través del avance automático de las diapositivas. Sin embargo, en el momento de la explicación por parte del profesor, los estudiantes manifestaron la necesidad de que el docente pudiera marcar los ele-

mentos en la transparencia con algún puntero virtual. Esto agilizaría la búsqueda de los objetos en las diapositivas y evitaría que los alumnos se desconcentren.

Los resultados de la encuesta y las estadísticas que se pueden obtener de los registros de seguimiento de actividades ponen de manifiesto que la metodología es aceptada y valorada tanto por los alumnos como por los docentes. La interacción virtual sincrónica permanente entre los participantes permitió redimensionar lo social, mejorar la retroalimentación y dar más dinamismo al curso. La continuidad en las tareas de enseñanza evitó la sobrecarga horaria durante períodos cortos. Asimismo, permitió que los alumnos tuvieran intervalos de tiempo suficiente entre clase y clase, para lograr la maduración de los conocimientos, habilitando al profesor, en las clases subsiguientes, a profundizar en la temática y a comparar el tema expuesto con los anteriores.

2.2.3. Adopción de nuevas tecnologías de enseñanza en la Universidad

Introducción

Este caso estudia la adopción de la herramienta *Horizon Wimba Virtual Classroom* en cursos en línea en una universidad del sureste de los Estados Unidos. El propósito es explorar los distintos factores de adopción (organizacional, social, personal y tecnológico) y las características del aula virtual encuestando a aquellos profesores que utilizaron el aula. En el otoño de 2010, los profesores fueron encuestados y entrevistados acerca de su decisión de adoptar y utilizar Wimba. El caso es una muestra de que la enseñanza utilizando comunicación sincrónica (y contando con las herramientas y profesionales adecuados) puede ser aplicada en cursos universitarios.

Motivación

Los maestros usan tecnologías cuando se dan cuenta de que motivan el aprendizaje de los estudiantes y mejoran la instrucción (Ajjan & Hartshorne, 2008). Para que la adopción de la tecnología tenga éxito, los instructores deben estar dispuestos a cambiar su papel en el aula, de ser un maestro a un facilitador y los estudiantes deben asumir la responsabilidad de su aprendizaje.

La integración exitosa de la tecnología en el aula depende de la alteración simultánea de las creencias y la filosofía de los maestros. Factores personales como la actitud, el interés y el

entrenamiento son relevantes para los profesores que adoptan tecnología y aquellos profesores que confían en su capacidad para manejar herramientas tecnológicas tendrán más probabilidades de integrar estas herramientas en su enseñanza (Lin & Poirot, 2009).

Herramienta utilizada

Al igual que otras aulas virtuales y software de conferencia, Horizon Wimba es un entorno virtual con audio, video, uso compartido de aplicaciones y visualización de contenido (Figura 2.3). Las características del aula virtual de Horizon Wimba se agrupan en tres categorías según su aplicación:

- La discusión y la interacción en las salas mediante el uso de emoticones, chats, videos, presentaciones, encuestas y cuestionarios.
- La instrucción y el refuerzo a través de la pizarra electrónica, el uso compartido de aplicaciones y el área de contenido.
- Las herramientas de gestión, que incluyen la capacidad de cargar y almacenar documentos, una lista de participantes y detalles de uso.

Metodología

Este estudio de caso se llevó a cabo en la Universidad de Carolina del Norte en Wilmington, que utilizó Horizon Wimba durante cinco años. En ese momento la universidad tenía 602 profesores de tiempo completo y 288 de tiempo parcial. Los profesores poseían acceso al aula virtual de Wimba. En el otoño de 2010, se realizó una encuesta usando una herramienta en línea. La oficina de *e-learning* envió un correo electrónico con un enlace a la encuesta y un breve mensaje sobre su propósito a 52 profesores identificados como usuarios de Wimba. Se informó a la universidad de que su participación era voluntaria y anónima.

La encuesta estuvo disponible por un período de tres semanas y durante este tiempo se envió un recordatorio de correo electrónico. Veintitrés profesores completaron la encuesta, lo que resultó en una tasa de respuesta del 44 %. Se utilizaron estadísticas descriptivas para informar los datos de la encuesta. Al completar la encuesta, se pidió a los profesores que proporcionaran información de contacto si estaban dispuestos a participar en una entrevista de seguimiento. Diez instructores expresaron interés y debido a la disponibilidad, seis fueron entrevistados varias semanas más tarde. Cada entrevista se realizó dentro del aula virtual de Wimba, donde

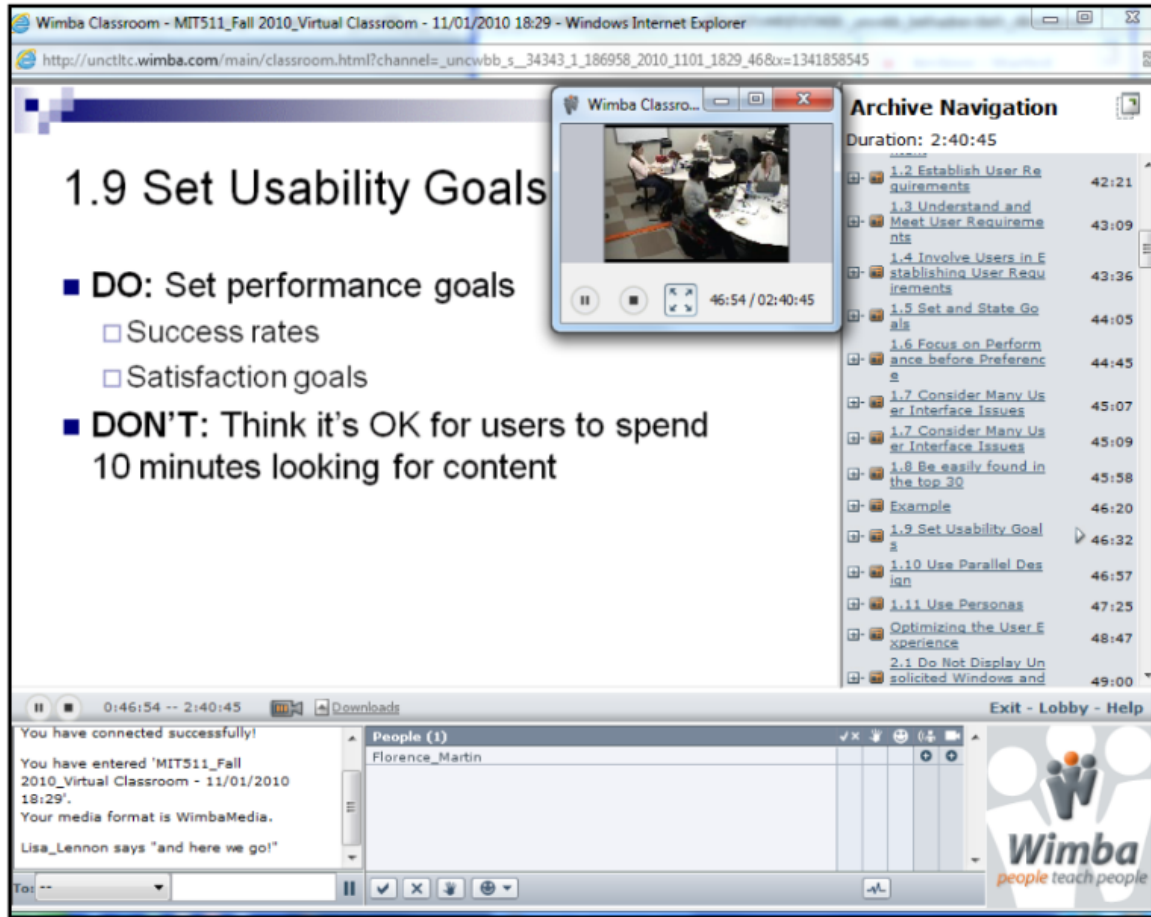


Figura 2.3: El aula virtual de Horizon Wimba.

los instructores podían ver las características según describían su uso. Las entrevistas fueron grabadas en Wimba y luego transcritas.

Resultados

Los docentes destacaron que Wimba **promueve la presencia social entre estudiantes** (61,8%) y **promueve el sentido de comunidad** (59,2%) fueron altamente calificados entre los factores sociales. **Mejorar la enseñanza** (66%) y **mejorar el aprendizaje del estudiante** (73%) tuvieron el promedio más alto entre los factores personales. La **facilidad de configuración** (74,8%) y la **disponibilidad de tecnología** (78,2%) tuvieron el mayor puntaje al considerar factores tecnológicos.

Los encuestados calificaron la herramienta como **ventajosa en comparación con otras metodologías de enseñanza** (60,8%) y **compatible con sus valores, necesidades y experiencias** (60%). Estos ítems fueron clasificados como superiores a las otras características de

la innovación como **simplicidad de uso** y **disponibilidad para la experimentación**.

Según Rogers (1995), los individuos en un sistema social no adoptan una innovación al mismo tiempo. Un cierto porcentaje de individuos es propenso a adoptar más temprano o más tarde una nueva idea. Basado en el criterio de la innovación, la mayoría de los participantes se calificaron como **innovadores** (39,1 %), seguidos por una gran cantidad de **primeros adoptantes** (34,8 %), luego aquellos que adoptaron la innovación **después de un tiempo** (21,7 %) y solo uno de los participantes (4,3 %) adoptó la innovación **luego del promedio de tiempo** que el resto.

Respecto de como utilizaron los profesores las posibilidades del aula de Wimba, la función de **grabar la sesión** es la característica más frecuentemente utilizada (61,8 %), seguida por el **chat de audio** (52,2 %), **levantar la mano** (51,4 %) y el **chat de texto** (49,6 %). Mientras que el **acceso como invitado** (14,8 %) y la **descarga del archivo como MP3** (15,6 %) fueron los menos utilizados por unanimidad.

Rogers también sostiene que hay 5 parámetros que influyen la decisión de un individuo a la hora de adoptar o rechazar una innovación. Estos parámetros son la relativa ventaja, compatibilidad, complejidad, disponibilidad y observabilidad.

Ventaja relativa se refiere a las ventajas de la innovación sobre la generación anterior, en este caso, la forma tradicional de enseñanza cara a cara. Los profesores pueden utilizar el aula virtual no sólo en los cursos en línea, sino también en sus cursos combinados, para agregar flexibilidad en el desarrollo de los cursos y pueden ofrecer horas de oficina virtual. Los estudiantes pueden participar en las clases y colaborar con sus compañeros sin tener que asistir al campus. Otras ventajas son la mejora en la interacción alumno-alumno y profesor-alumno, el permitir enfoques de enseñanza centrados en el estudiante y la oportunidad de proporcionar retroalimentación inmediata a los estudiantes.

La **compatibilidad** es el nivel por el cual la innovación tiene que ser asimilada en la vida de un individuo. La universidad consideró que Wimba era compatible con otros métodos de enseñanza. Los profesores son capaces de replicar estrategias de enseñanza similares de un aula cara a cara, llamar a los estudiantes, obtener respuestas de ellos de inmediato, y también hacer actividades de grupo con ellos. Además, la herramienta pudo ser integrada con el sistema existente de gestión de cursos que utilizaba la universidad.

La **complejidad** es la probabilidad de que una innovación sea adoptada por un individuo en base a lo difícil que es utilizarla. La facultad consideró la facilidad de uso y la facilidad de instalación como factores importantes para la adopción del aula virtual. La facultad pudo utilizar

esta herramienta con un entrenamiento mínimo; la interfaz era lo suficientemente institucional para que tanto los estudiantes como los profesores aprendieran las diferentes características. Se observó que los estudiantes se sentían cómodos navegando la herramienta desde el período de la segunda clase.

La **disponibilidad** determina con que facilidad se puede acceder a experimentar una innovación. En este caso, no hubo restricciones en el número de salas que cada facultad podía usar, ni en el tiempo de uso, ni en la cantidad de personas utilizando simultáneamente las salas. Además, debido a las grabaciones, los usuarios pudieron ver las conferencias a las que no pudieron asistir.

La **observabilidad** es la medida en que una innovación es visible para los demás. Una innovación que sea más visible, impulsará la comunicación entre los pares del individuo y sus redes personales, creando así más reacciones positivas o negativas. En la universidad, sólo alrededor del 10 % utilizaba la herramienta de aula virtual en sus cursos; a pesar de su fomento por parte de la oficina de *e-learning*, aun no había una visibilidad en todos los niveles para usar la herramienta. Esta característica de la innovación fue calificada como la más baja entre las cinco.

En este caso de estudio, aunque ninguna de estas características fueron clasificadas como extremadamente altas, todas fueron clasificadas cerca del 60 %.

2.3. Conclusión

En estas experiencias se observó que se utilizan herramientas ad-hoc en cada caso y que tienen en común la posibilidad de solicitar la participación, realizar encuestas, grabar videos, etc. Las experiencias se enfocaron en aspectos como alcanzar prácticas comunicativas, grabar sesiones de trabajo para dejarlas disponibles para todos y dar la idea de comunidad.

Al repasar estos antecedentes, vemos como ya desde hace tiempo tanto alumnos como docentes, mostraban entusiasmo en la adopción de estas herramientas. Hoy por hoy, es innegable que el aporte de las herramientas sincrónicas enriquece notablemente la experiencia educativa en el desarrollo de un curso.

Gracias al paso del tiempo, algunos problemas como la calidad del sonido y del video o la desconexión de la sesión, han mejorado notablemente debido a la evolución del hardware que da soporte a estas herramientas.

Por otro lado, también es predecible que con el tiempo, la resistencia a la adopción o falta de entusiasmo en algunos docentes, debido a la disrupción que genera el nuevo enfoque digital, se vea disminuida. Se espera entonces, un incremento en el uso de estas herramientas durante los próximos años, debido a que las nuevas generaciones de docentes son más propensas a la adopción de nuevas tecnologías.

Capítulo 3

Herramientas de comunicación sincrónica

3.1. Introducción

En este capítulo se analizarán en detalle algunas de las herramientas de comunicación sincrónica utilizadas en la actualidad.

Se comienza mencionando algunos antecedentes históricos y definiendo una serie de criterios que serán utilizados como base para la comparación de las herramientas. Esta selección de criterios, permitirá descubrir tanto las características en común de los software analizados, así como también como los aspectos en los que se diferencian y determinan su aptitud en un dominio determinado.

Luego se realiza el análisis detallado de las tres herramientas elegidas, recorriendo cada aspecto antes mencionado.

Finalmente, se concluye el capítulo destacando las características más negativas y más sobresalientes de las herramientas.

3.1.1. Selección de herramientas y aspectos a comparar

Al buscar antecedentes de herramientas de comunicación, una de las primeras herramientas que se encuentra es Lyceum, desarrollada en la Open University (OU) del Reino Unido en el año 2002. Lyceum permitía que un grupo de usuarios se comuniquen verbalmente por internet usando voz sobre IP. Además ofrecía un chat para que los usuarios puedan comunicarse por escrito y una pizarra interactiva para escribir y compartir imágenes.

El buen impacto que tuvo esta nueva forma de interacción entre los usuarios, dio lugar al desarrollo de nuevos programas. Muchos de ellos fueron descontinuados como Yugma¹ o WebHuddle².

Otras herramientas tuvieron más suerte y fueron adquiridas por grandes empresas, como el caso de WebEx que fue adquirida por Cisco³ o el caso de DimDim, quien fue adquirida por Salesforce⁴.

Al día de hoy, existen una gran variedad de herramientas, donde la gran mayoría son plataformas web, debido al declive del software de escritorio y la naturaleza distribuida de la web que la convierte en una plataforma adecuada para este tipo de software. Dentro de estas herramientas encontramos dos grupos. Por un lado, el grupo de herramientas *open source*, donde se encontró que las más destacables por su funcionalidad, documentación y popularidad son BigBlueButton y Apache Open Meetings. Y por otro lado, las herramientas comerciales, bajo el modelo de servicio en la nube, en el que se considera a Cisco WebEx como la más completa.

A continuación, se proponen una serie de características presentes en las herramientas de comunicación estudiadas, las cuales servirán como base para hacer una comparación de las mismas, así como también para definir algunos de los requerimientos de InfoMeeting.

- Acerca de la plataforma: se analizarán aspectos como la cantidad de usuarios soportados, los sistemas operativos en los que se pueden ejecutar las herramientas, la facilidad de instalación tanto para los usuarios finales como para los administradores y el tipo de licencia que tiene el software.
- Modos de interacción: se verán las distintas maneras que tienen los usuarios de interactuar con el software ya sea mediante audio, video, compartiendo archivos, completando encuestas y/o mediante el uso de chats y mensajería privada.

¹<https://www.yugma.com/>

²<https://www.webhuddle.com/>

³Trask, Amy (October 14, 2002). "File Bothell's LapLink under 'S' for survivor". The Seattle Times.

⁴Agrawal, Shrija (November 3, 2010). "Salesforce.com Close To Acquiring Web Collaboration Company Dimdim". VC Circle.

- Administración y sesiones: desde el punto de vista del administrador se analiza la gestión de los usuarios, las sesiones y la posibilidad de agendar y persistir las mismas.
- Experiencia de usuario: se analiza la curva de aprendizaje de la herramienta para los usuarios finales, el diseño de la interfaz y la facilidad para cargar y guardar contenido.
- Integración: se verá cómo se integran con otros sistemas, ya sea mediante el uso de *plugins*, APIs o protocolos de autenticación.
- Otras características: se destacan otros aspectos de las herramientas.

3.2. Apache Open Meetings

Open Meetings es una aplicación de conferencia web de código abierto que se puede utilizar libremente bajo los términos de la licencia GNU. Siendo de código abierto, la aplicación no es perfecta, pero recibe actualizaciones constantemente y cualquier persona es libre de desarrollar nuevas características para ello. También existen algunas compañías que ofrecen servicios para AOM, incluyendo soporte de código, desarrollo, integración y corrección de errores.

3.2.1. Acerca de la plataforma

Cantidad de usuarios

Las sesiones de AOM soportan entre 1 y 150 usuarios

Sistemas operativos soportados

La instalación de un servidor AOM puede hacerse tanto en sistemas operativos Linux, como en Mac OSX y Windows.

Soporte en plataformas mobile

AOM no cuenta con aplicaciones nativas oficiales para acceder a las sesiones desde dispositivos móviles. Existe un cliente para el sistema operativo Android llamado OmMobile⁵ el cual

⁵<https://play.google.com/store/apps/details?id=air.org.apache.openmeetings.OmMobile>

no es gratuito y carece de muchas de las características que presenta la versión web. Sin embargo, es posible acceder a la aplicación desde el navegador de un dispositivo móvil, aunque la experiencia se ve limitada por la ausencia de un diseño adaptable a pantallas pequeñas. Otro factor limitante, puede darse en la falta de compatibilidad entre los *plugins* requeridos por la herramienta con los que posee el navegador del móvil.

Escalabilidad

OpenMeetings permite instalarse en múltiples servidores. Para ello, todas las instancias deben compartir la misma base de datos y tener configurada la misma zona horaria para evitar problemas de expiración con las sesiones de los usuarios. Los servidores puede agregarse y configurarse desde el panel de administración. Por otro lado, es necesario compartir ciertas carpetas entre los servidores para poder acceder a los archivos subidos o a las sesiones grabadas.

Facilidad de instalación (cliente)

El usuario final no necesita ningún software especial, solo debe tener un navegador moderno que cuente con los *plugins* adecuados para la reproducción de audio y video.

Facilidad de instalación (servidor)

Para instalar OpenMeetings hay que dirigirse a la sección descargas de su sitio web⁶ y descargar la versión que se desea instalar, siendo 3.1.3 la última versión disponible a la fecha. Luego hay que extraer los contenidos del archivo descargado y ejecutar el instalable red5 (red5.sh para sistemas unix o red5.bat para windows). Por último hay que abrir el navegador en la URL <http://localhost:5080/openmeetings/install> y seguir los siguientes pasos:

1. Completar la información para conectar a la base de datos
2. Establecer el nombre, correo electrónico y contraseña del usuario administrador
3. Configurar los parámetros básicos del sistema (ej: permitir que los usuarios se auto-registren)

⁶<http://openmeetings.apache.org/downloads.html>

4. Configurar las rutas de las herramientas auxiliares (servidor de mail, *codecs* de audio y video, conversor de PDF, etc).
5. Definir el mecanismo de encriptación de datos

Una vez completada la información requerida, la aplicación quedará instalada y lista para usarse. Por defecto AOM usa una base de datos integrada llamada Apache Derby. Para un ambiente de producción se sugiere usar una base de datos como MySQL o Postgres. Si se pretende usar la pizarra interactiva para subir e importar documentos, es necesario instalar las siguientes herramientas en el servidor:

- *ImageMagik* para el soporte de imágenes
- *GhostScript* y *SWFTools* para el soporte de PDFs
- *JODConverder* y *OpenOffice/LibreOffice* para documentos *office* (DOC, XLS, PPT)
- *FFMpeg* y *SoX* para importar video. Estas herramientas también son necesarias para grabar sesiones.

Licencia

El uso de la plataforma se encuentra bajo la Licencia Apache

3.2.2. Modos de interacción

Soporte de audio y video

AOM tiene soporte para audio y video. Al momento de crear una sala, se presenta la opción de elegir entre ambas modalidades o la opción de sólo audio. Cuando un participante se une a una sesión, puede elegir entre cuatro modalidades:

1. Utilizar audio y video
2. Utilizar sólo audio
3. Utilizar sólo video
4. No utilizar ni audio ni video (imagen estática)

En esta instancia, también es posible elegir los dispositivos a utilizar. En el caso de la cámara, se puede elegir la resolución que se desee utilizar en la sesión y se permite hacer un test de grabación de 5 segundos.

Encuestas

Durante una conferencia, el usuario moderador puede crear encuestas. Estas encuestas llevan un título y pueden ser de dos tipos: *por sí o por no* y *numéricas* (hasta 10 opciones). El resto de los usuarios, al igual que el moderador, tienen la posibilidad de responder a la encuesta. En el caso del moderador, tendrá que ir al menú de acciones y seleccionar la opción de "Votar", mientras que para el resto de los participantes se presentará una ventana emergente dándole la opción de votar.

Desde el menú de acciones, la opción *poll results*, permite elegir alguna de las encuestas de la sesión y muestra los resultados de las mismas, pudiendo elegir la visualización a través de un gráfico de tortas o un gráfico de barras. Cabe destacar que cuando se crea una nueva encuesta, sólo es posible votar en esta encuesta recién creada y ya no se podrá votar en alguna de las anteriores que haya quedado sin responder.

Chat

Además de audio y video, las salas tienen un soporte para chat. El chat puede ser tanto público (entre todos los usuarios) como privado. Entre las opciones del chat, se puede elegir el color y el estilo (negrita, cursiva o subrayado) de la fuente. Cuenta también con soporte para emoticones. Otra de las opciones es ver un *log* de todos los mensajes. Desde esta vista es posible seleccionar el texto para copiarlo a un archivo y guardarlo. Por otro lado, el usuario administrador tiene la posibilidad de borrar todo el historial de mensajes.

Pizarra

La herramienta posee una pizarra o *whiteboard* (Figura 3.1) bastante completa que permite:

1. Escribir y borrar texto
2. Dibujar formas geométricas

3. Dibujar libremente
4. Seleccionar objetos
5. Deshacer y guardar cambios
6. Establecer el nivel de zoom

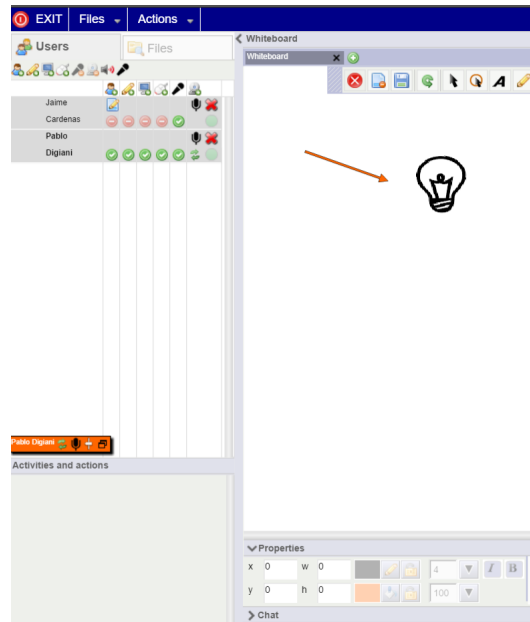


Figura 3.1: Pizarra y panel de usuarios en AOM. Los moderadores pueden otorgar permisos a los demas usuarios para compartir pantalla, usar el whiteboard, camara y micrófono.

Compartir archivos

AOM permite compartir en la pizarra documentos del tipo PPT, DOC, XML e imágenes.

Compartir pantalla

La herramienta provee una función para compartir pantalla, la cual se ejecuta a través de una aplicación java que se ofrece para descargar en el momento que el usuario decide compartir la pantalla. La aplicación permite configurar el área de la pantalla a compartir, y, una vez que el usuario decide comenzar a compartir, los demás participantes recibirán una notificación donde pueden aceptar o rechazar la invitación a ver la pantalla.

3.2.3. Administración y sesiones

Grabado de sesiones

De la misma manera que se puede compartir la pantalla, es posible grabar parte de las sesiones (Figura 3.2), aunque para ello es necesario instalar previamente *ffmpeg* y *sox*. Una vez concluida la grabación, el administrador, puede ingresar al menú *Recordings* y ver la lista de grabaciones disponibles y descargarlas. También se mostrarán los eventuales errores que se hayan producido durante la creación de los archivos de video.

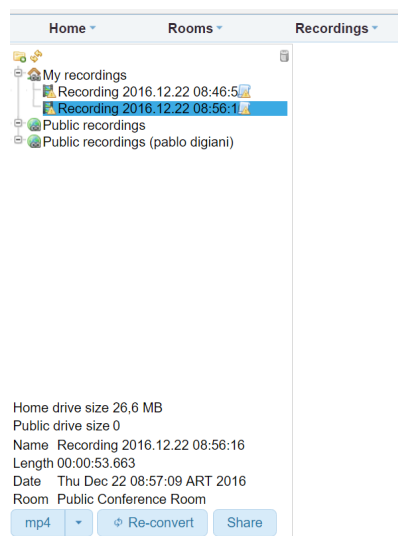


Figura 3.2: Sesiones grabadas en una instalacion de AOM

Administración de usuarios y salas

La creación tanto de usuarios como de salas, la realiza el usuario administrador desde el menú de configuración. Las salas pueden ser públicas, es decir que puede acceder cualquier usuario del sistema, o privados, donde el acceso se limita a determinados grupos de usuarios. Al momento de crear las salas, además de elegir el nivel de privacidad, se puede establecer la cantidad máxima de participantes, habilitar la opción de sólo audio, permitir el grabado de sesiones y configurar el diseño de la sala, pudiendo ocultar el chat, la pizarra, el menú de acciones y otros elementos visuales.

Calendario

AOM provee un calendario (Figura 3.3) con vista de mes, semana y día, donde se pueden ver los eventos agendados y agendar nuevos. Al crear un evento se debe ingresar la fecha y hora del mismo, los usuarios que participarán, y la sala donde se llevará a cabo el evento. También es posible configurar un recordatorio por correo electrónico a los participantes y proteger la entrada de la sala con una contraseña.

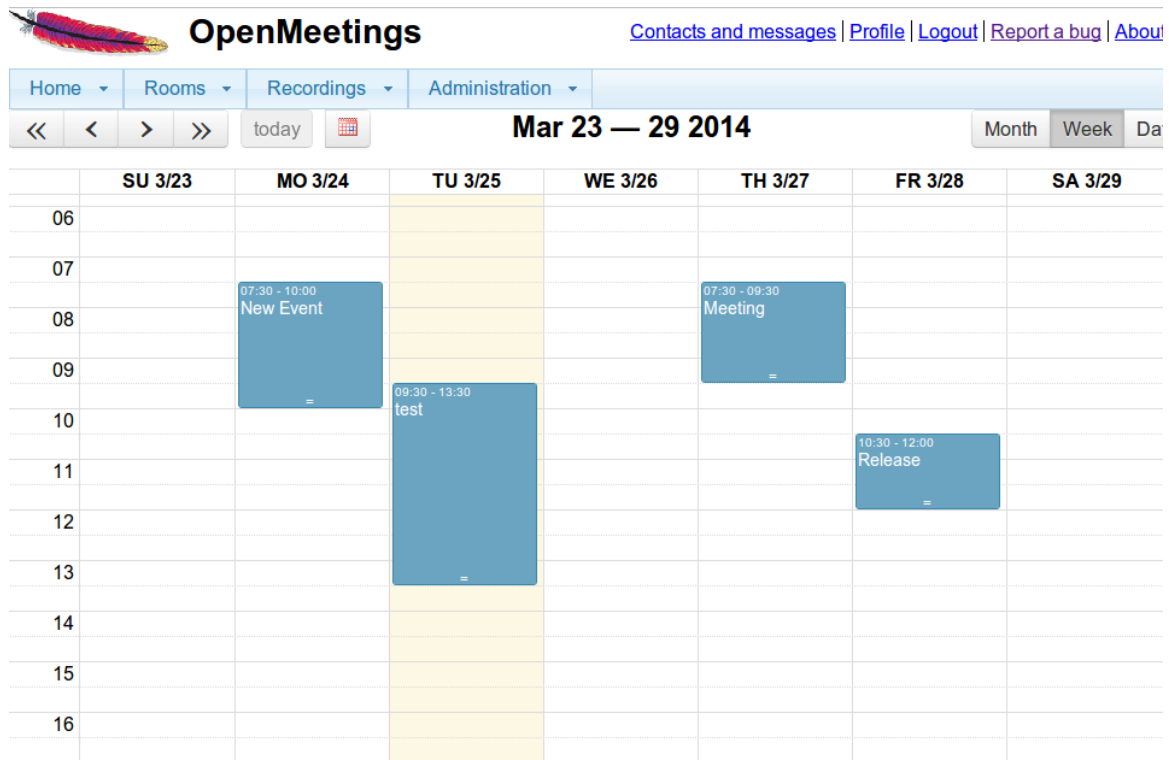


Figura 3.3: El calendario de AOM muestra los eventos agendados

Sesiones públicas

AOM no dispone de sesiones públicas y sólo los usuarios registrados en el sistema pueden asistir a las sesiones.

3.2.4. Respeto de la Experiencia de Usuario

Curva de aprendizaje

El menú de administración resulta muy intuitivo, ya que, una vez instalado, es posible cargar rápidamente algunas salas y usuarios para comenzar a probar el sistema. En cuanto al uso de las salas, también resulta intuitivo comenzar a interactuar tanto por parte de los usuarios como por parte del moderador. La grilla de usuarios está muy bien organizada y facilita la tarea del moderador, dado que en poco espacio puede ver la lista de participantes y efectuar varias acciones sobre ellos como: silenciarlos, permitirles compartir la pantalla o permitirles usar la pizarra.

Interfaz gráfica

La interfaz está bien diseñada, aunque los detalles visuales como los íconos o los botones se ven un tanto anticuados en comparación con otras interfaces web más modernas. Tanto las salas como el panel de administración tienen un diseño adaptable a resoluciones bajas.

Configuración inicial y backups

Desde el menú de administración es posible exportar e importar datos mediante una copia de seguridad del sistema. Estas copias o *backups* incluyen todos los datos generados por los usuarios, excepto la configuración que se define al instalar el sistema. Además del menú de administración, otra manera de importar y exportar datos, es desde la terminal mediante los comandos *backup* y *restore*. Esta opción es mas rápida y recomendable cuando el tamaño del *backup* es muy grande.

3.2.5. Integración con otros sistemas

AOM cuenta con una gran variedad de formas a la hora de interactuar con otros sistemas

API

Cuenta con APIs REST y SOAP desde las cuales se pueden obtener y crear entidades como: eventos del calendario, usuarios, grupos, salas, grabaciones, archivos e incluso instancias de

servidores.

LDAP

Open Meeting posee además, integración con el protocolo LDAP, para obtener y utilizar los usuarios de Active Directory.

OAuth2

También es posible integrar OpenMeetings con el protocolo OAuth2. Servicios como Facebook y Google ya vienen preconfigurados. Completando los datos necesarios, el administrador puede agregar otros servicios que implementen OAuth2.

Plugins

AOM cuenta con una gran variedad de *plugins* para crear y acceder a las salas desde otros sistemas conocidos. Los sistemas más conocidos que poseen esta integración son: Moodle, Jira, y Redmine, entre otros.

3.2.6. Otras características

Múltiples Idiomas

Está disponible en varios idiomas como: inglés, francés, ruso, español y japonés entre otros. Además posee un editor de etiquetas, en caso de que el administrador quiera modificar el texto de algún elemento visual.

Herramienta de línea de comandos

AOM trae incorporada una herramienta que permite ejecutar comandos para realizar tareas como:

- Realizar una instalación con la configuración por defecto
- Realizar una instalación a partir de un *backup*

- Realizar un *backup* del sistema
- Ejecutar una limpieza (*cleanup*) del sistema

Mensajería privada

AOM permite enviar mensajes a los usuarios y organizarlos en carpetas. Además, es posible confirmar la asistencia a un evento mediante el uso de mensajería privada.

3.3. BigBlueButton

BigBlueButton es un software de conferencia web de código abierto, basado en el sistema operativo GNU / Linux y se ejecuta en Ubuntu. Fue creado por BigBlueButton Inc., una compañía sin fines de lucro constituida en Ontario, Canadá, que posee los derechos de autor del código fuente y la marca comercial de BigBlueButton.

3.3.1. Acerca de la plataforma

Cantidad de usuarios

BBB soporta sesiones de hasta 80 usuarios.

Sistemas operativos soportados

La versión 1.0 de BigBlueButton fue diseñada, desarrollada y probada en Ubuntu 14.04 64-bit. El fabricante, no hizo pruebas en otras versiones de Ubuntu. La versión 1.1, aún en desarrollo, está siendo preparada para la versión más reciente de Ubuntu (16.04).

Soporte en plataformas mobile

No hay aplicaciones nativas para acceder a BBB. Se puede acceder vía web, aunque la experiencia se ve limitada por la ausencia de un diseño adaptable y por la falta de compatibilidad entre los *plugins* requeridos por la herramienta y los del navegador móvil.

Escalabilidad

Si bien no es posible desplegar BBB en varios servidores, al menos se puede, editando un archivo de configuración XML, separar las funciones de audio, pantalla compartida y video en servidores diferentes y utilizar un servidor en común para gestionar las sesiones.

Facilidad de instalación (cliente)

Sólo necesita un navegador moderno con *plugins* para audio y video.

Facilidad de instalación (servidor)

Para instalar BBB se deben cumplir ciertos requisitos como:

- Un sistema operativo Ubuntu 14.04 64-bit
- El puerto 80 libre
- 4 GB de memoria (aunque se recomiendan 8 GB)
- Al menos un procesador Quad-core 2.6 GHZ

Dado que muy pocos servidores compartidos cumplirían tales requisitos, es necesario montar uno exclusivamente para desplegar la plataforma. El proceso de instalación consiste en descargar los siguientes paquetes:

- *PPA* para ver documentos compartidos
- *ffmpeg* para reproducir audio, video y compartir pantalla
- *bigbluebutton*, el paquete principal de la plataforma

La instalación de dichos paquetes incluye un servidor web Nginx y el media server Red5.

Licencia

El uso de la plataforma se encuentra legislado bajo las licencias GPL y LGPL

3.3.2. Modos de interacción

Soporte de audio y video

BBB soporta tanto audio como video. Al entrar a una sesión (Figura 3.4), se presenta la opción de elegir entre usar el micrófono o participar como "solo escucha". En cuanto al video, se ofrece antes de comenzar a transmitir, la posibilidad de elegir la calidad y la cámara web a utilizar.

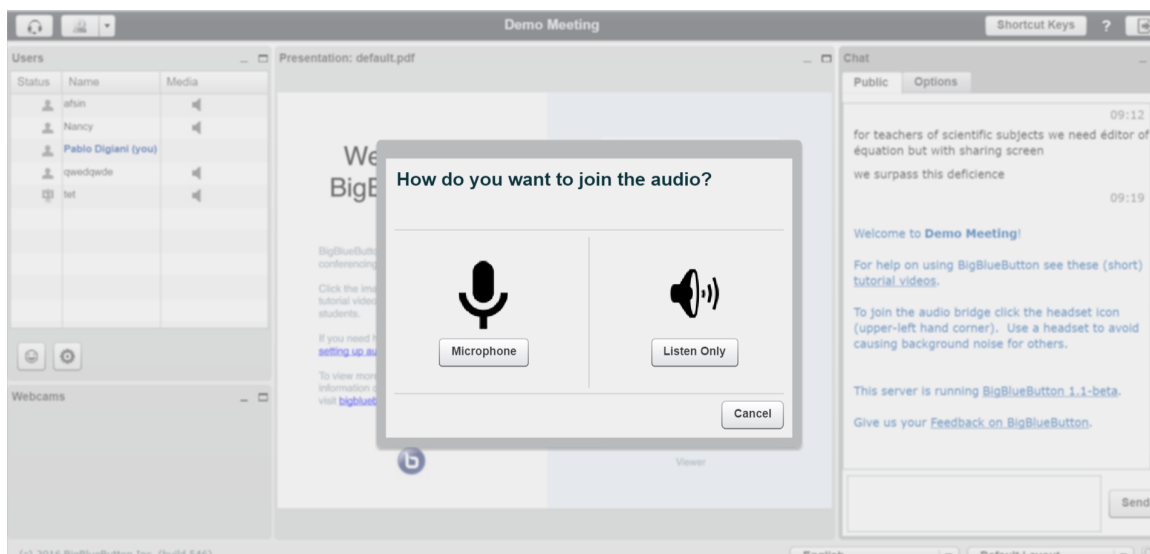


Figura 3.4: Al momento de entrar a una sesión podemos elegir entre dos modos de participación

Encuestas

Desde la pizarra, se pueden iniciar encuestas del tipo si/no, verdadero/falso, o de múltiples opciones. A la hora de votar, la consigna de las encuestas no es incluida. Esto obliga al usuario que inicia la encuesta, a indicarle al resto de los participantes el propósito la misma, ya sea usando audio o el chat general.

Chat

BBB tiene soporte de chat tanto a nivel grupal como individual. Entre las opciones del chat se puede elegir el tamaño de la fuente y la posibilidad de activar notificaciones de audio cuando se recibe un mensaje privado (Figura 3.5).



Figura 3.5: BBB permite enviar mensajes de chat privados

Pizarra

La pizarra es bastante completa y entre sus opciones permite:

- Hacer movimientos (*pan*) y *zoom*
- Escribir libremente, eligiendo el grosor del trazo y el color
- Dibujar formas geométricas e insertar texto
- Borrar o deshacer cambios

Compartir archivos

BBB permite subir y mostrar archivos usando la pizarra (Figura 3.6). Tiene soporte para archivos del tipo PDF, PPT, DOC, XLS e imágenes. La pizarra incluye controles para avanzar entre páginas cuando se visualiza un archivo.

Compartir pantalla

El moderador puede compartir la pantalla descargando un archivo *jnlp*. Una vez abierto puede elegir la región a compartir (o pantalla completa). Los usuarios que ven la pantalla compartida pueden verla en el tamaño original o bien pueden hacer un ajuste del contenido en la ventana donde se muestra la pantalla compartida.

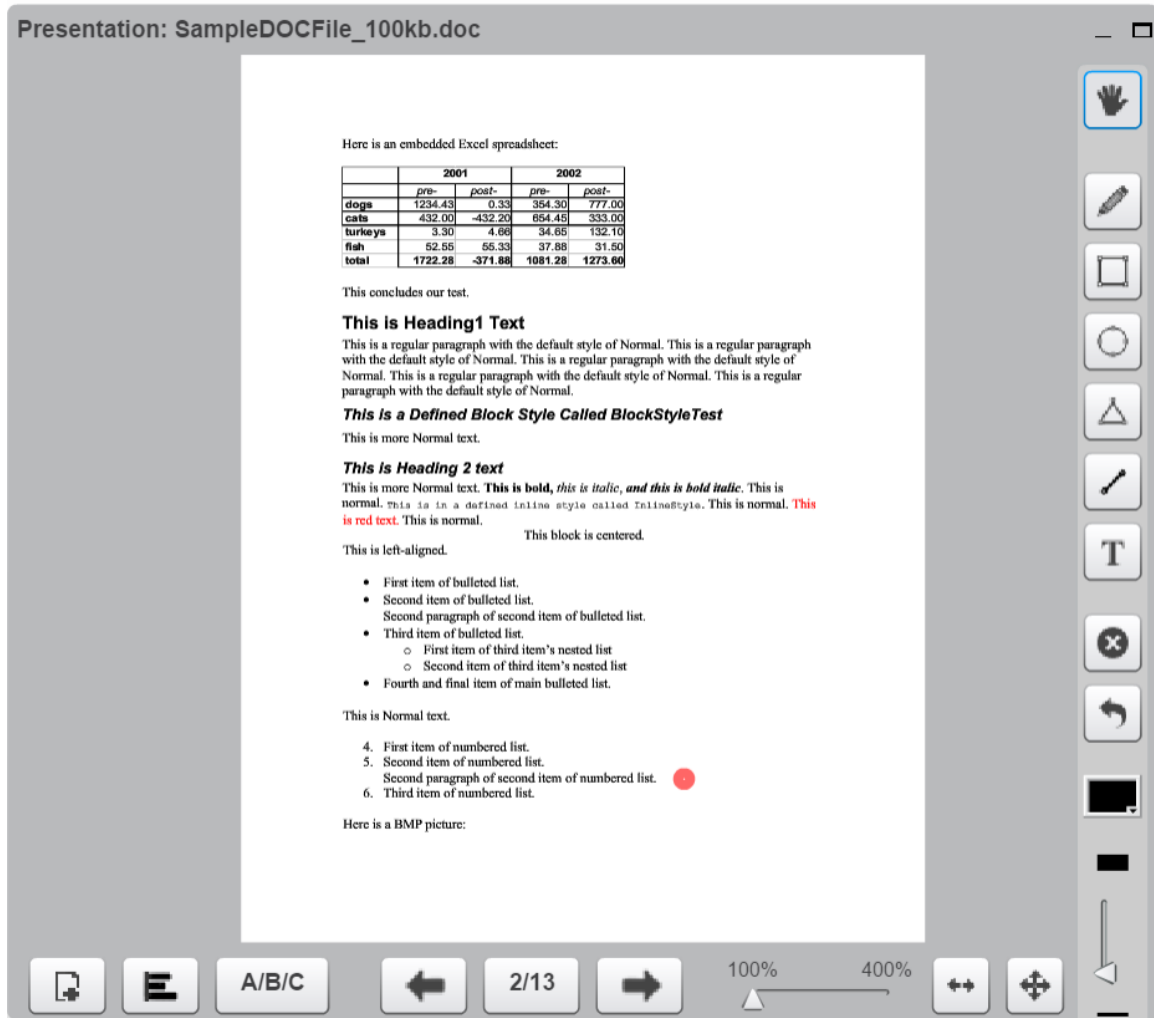


Figura 3.6: Pizarra o *Whiteboard* de BBB, compartiendo un archivo .doc

3.3.3. Administración y sesiones

Grabado de sesiones

BBB permite el grabado de sesiones utilizando software de otros fabricantes que interactúen con su API. Cuando iniciamos una sesión con el parámetro *'record=true'*, aparecerá una nueva opción en la barra de herramientas, la cual permite comenzar y pausar el grabado de la sesión.

BBB guardará los archivos subidos, los chats, audios, webcams, screen shares y eventos de la pizarra. De esta manera, una vez finalizada la sesión, se ejecutarán los scripts necesarios para procesar los datos capturados y convertirlos en el formato de video adecuado para su reproducción. BBB no cuenta con una interfaz gráfica para administrar las sesiones grabadas, pero provee una herramienta de línea de comandos llamada *bbb-record* que permite listar, ver

y borrar las grabaciones generadas.

Administración de usuarios y salas

En sus primeras versiones, BBB comenzó a desarrollar una interfaz de administración para crear cuentas, programar sesiones, configurar las propiedades de las sesiones y otros aspectos. Sin embargo, el desarrollo fue abandonado dado que la mayoría de las instalaciones de BBB presentaban las siguientes características:

- Se integraban con otros software open source como Joomla o WordPress, con lo cual los administradores optaban por administrar todo desde estos otros sistemas.
- Se integraban con una aplicación web, con la cual los administradores necesitaban una API para controlar la sesiones.
- Simplemente querían tener una sesión lista para usar sin necesidad de configuraciones

Debido a esos motivos, la administración de sesiones y usuarios se realiza programáticamente editando archivos jsp o a través de la API.

Calendario

BBB no posee un calendario. La única manera de integrar un calendario es integrando BBB con algún sistema de gestión de contenido que provea tal característica, o bien desarrollar un módulo para ello.

Meetings públicas

Las sesiones son públicas por defecto. Al configurar una sesión, BBB nos da la posibilidad de elegir un password a nivel administrador y otro a nivel participante para proteger la entrada a la sala.

3.3.4. Respeto de la Experiencia de Usuario

Curva de aprendizaje

Se entiende que por el lado de los usuarios la curva de aprendizaje es rápida, gracias a una interfaz intuitiva y dividida en módulos (Figura 3.7). Cada módulo realiza tareas específicas, como el módulo de chat, la pizarra o las pantallas compartidas. Estos se muestran en ventanas independientes, las cuales pueden ser minimizadas por el usuario en caso de que no le sean útiles. Por otro lado, la aplicación dispone de siete *layouts* predefinidos, para mejorar la experiencia según el tipo de actividad que se quiera realizar (video chat, presentación de documentos, lectura, etc)

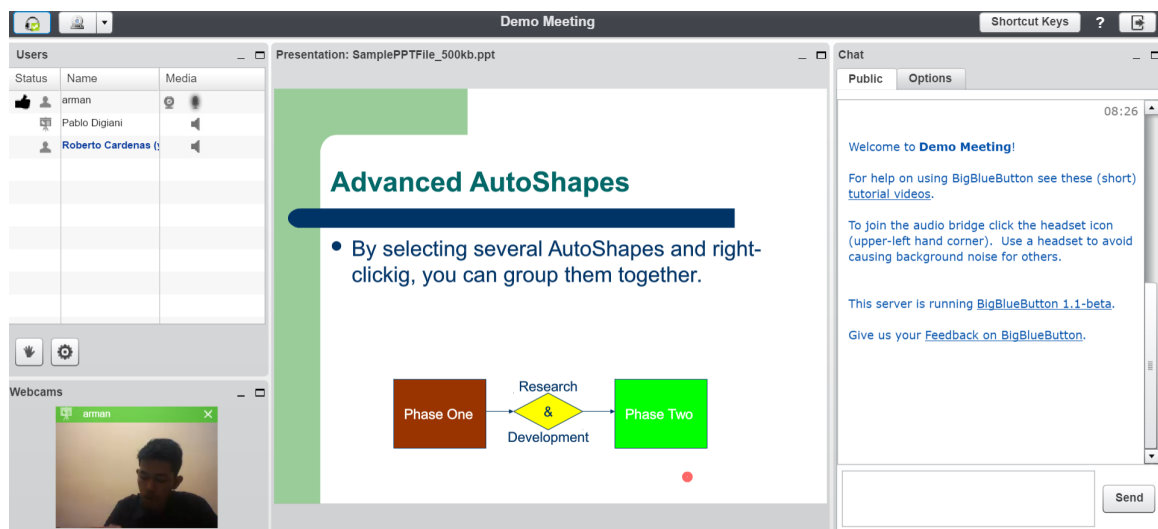


Figura 3.7: Pantalla principal de BigBlueButton: a la izquierda se observa el panel de usuarios con tres participantes conectados, en el centro se muestra un documento ppt compartido y a la derecha se encuentra el panel de chat.

Cabe destacar además, que al entrar a un salón, la pantalla recibe al participante con un documento cargado, que incluye enlaces a videos donde se explica cómo usar la herramienta. Por el lado de los administradores, la experiencia es totalmente distinta, ya que al no contar con una interfaz gráfica para gestionar las sesiones la curva de aprendizaje depende de qué tan familiarizado esté el administrador para escribir código.

Interfaz gráfica

La interfaz de usuario está muy bien realizada, con tamaños de fuente legibles y una iconografía clara e intuitiva. Además de traer diseños predefinidos, el sistema de ventanas permite

arrastrar, minimizar y maximizar los distintos módulos dándole al usuario la libertad de adaptar la interfaz a sus necesidades. Esto hace además, que se pueda utilizar el sistema en resoluciones bajas.

Configuración inicial y backups

Este es el aspecto más débil del sistema, ya que toda la importación de contenido deberá hacerse de forma programática debido que BBB está pensado para ser integrado a otros sistemas, en lugar de ser un sistema autónomo o *standalone*. El manual de usuario, indica cómo usar los comandos de Unix para poder transferir sesiones grabadas entre dos instalaciones.

3.3.5. Integración con otros sistemas

API

BBB posee una API web pensada para interactuar desde aplicaciones de terceros. Para efectuar una llamada a la API, el cliente debe conocer una clave, la cual debe enviar en cada llamado junto con un *checksum*. Estas medidas de seguridad permiten evitar ataques o llamados no deseados de otros clientes. La API posibilita a las aplicaciones crear y configurar sesiones, así como también monitorear el estado y obtener información de aquellas sesiones que se están llevando a cabo. También incluye funciones para gestionar las grabaciones realizadas.

Integración con otros productos open source

- Moodle: el módulo de actividades de Moodle permite crear, gestionar y unirse a sesiones de BBB desde Moodle. También soporta el grabado y reproducción de las sesiones.
- Sakai: es posible integrar una instancia de Sakai con un servidor BBB para crear, gestionar y unirse a las sesiones.
- WordPress: el *plugin* disponible permite a los administradores crear y gestionar las salas desde la interfaz de WordPress. Dicho *plugin* puede descargarse e instalarse desde el panel de administración de WordPress.
- Drupal: el módulo disponible implementa la API de BBB para que (de acuerdo a los roles que posean) los moderadores puedan crear nuevas sesiones y los usuarios puedan unirse a una.

- Otros proyectos *open source* menos conocidos que cuentan con integración de BBB son: Tiki Wiki CMS, eFront, RedMine, Foswiki, Atutor y Fedena.

3.3.6. Otras características

Multiples Idiomas

Gracias a los aportes de la comunidad BBB tiene traducciones en más de 20 idiomas.

Herramienta de línea de comandos

BBB cuenta con el comando *bbb-conf*, una herramienta de configuración que facilita la gestión de la configuración y el monitoreo de BBB, así como también la detección de potenciales problemas durante la instalación y configuración sistema.

3.4. WebEx

WebEx es una software que funciona bajo el modelo Software como un Servicio, abreviadamente ScuS (del inglés: *Software as a Service*, SaaS), un modelo de distribución de software donde la lógica y los datos que maneja el software se alojan en servidores de una compañía y se accede a ellos vía internet, mediante un navegador o un cliente específico. En este caso, la compañía Cisco dispone sus servidores en la llamada *Cisco Collaboration Cloud*, una red global de escala empresarial diseñada para la entrega segura de aplicaciones bajo demanda. Esta plataforma posee una arquitectura escalable, disponibilidad consistente y múltiples capas de seguridad, validada por rigurosas auditorías independientes, incluyendo SSAE-16 e ISO 27001. Por tal motivo, este análisis no contempla aspectos analizados previamente como facilidad de instalación, escalabilidad, licencia y *backups*.

3.4.1. Acerca de la plataforma

Cantidad de usuarios

La plataforma tiene una capacidad de hasta 1025 usuarios, aunque 25 de ellos pueden utilizar dispositivos de video en simultáneo.

Sistemas operativos soportados

Cisco WebEx funciona en todos los sistemas operativos comunes como Windows (desde Vista en adelante), MacOS a partir de su versión 10.7 y en la mayoría de las distribuciones Linux. En cuanto a los navegadores, WebEx soporta Internet Explorer 9 o mayor, Edge, Safari, Chrome y Firefox.

Soporte en plataformas móviles

Cisco WebEx puede ser accedido desde una gran variedad de dispositivos móviles. Es compatible con iPad, iPhone y dispositivos Android, Blackberry y Windows Phone 8.

3.4.2. Modos de interacción

Soporte de audio y video

WebEx cuenta con video de alta definición, utilizando una resolución de hasta 720 pixeles (1280x720). El vídeo de alta definición se desactiva automáticamente, cuando se pasa a un segundo plano debido al uso de compartido de aplicaciones o de escritorio.

En cuanto al audio, los participantes pueden optar por unirse por teléfono o por internet. El soporte de audio utilizando internet tiene una buena calidad de audio, incluso en redes de bajo ancho de banda. Los asistentes que se unen por teléfono muestran un icono de teléfono en el panel de asistentes, mientras que aquellos que lo hacen por internet tienen un icono de micrófono.

Encuestas

El organizador puede crear encuestas para reunir comentarios, probar conocimientos o votar durante la conferencia. Una vez completada la encuesta, los resultados de la misma se reciben al instante. El organizador tiene la opción de compartir los resultados con los asistentes y de guardarlos para su posterior análisis.

Chat

Los asistentes tienen la opción de chatear con otros, utilizando chat integrado en los paneles laterales. El mensaje puede ser enviado a un participante en particular o a todos los participantes.

Pizarra

El presentador puede escribir o dibujar en la pizarra de WebEx (Figura 3.8) y también puede asignar privilegios a otros participantes que pueden utilizar herramientas de texto y gráficos para agregar o resaltar notas e imágenes. Luego permite guardar y distribuir todos los documentos creados y anotados.

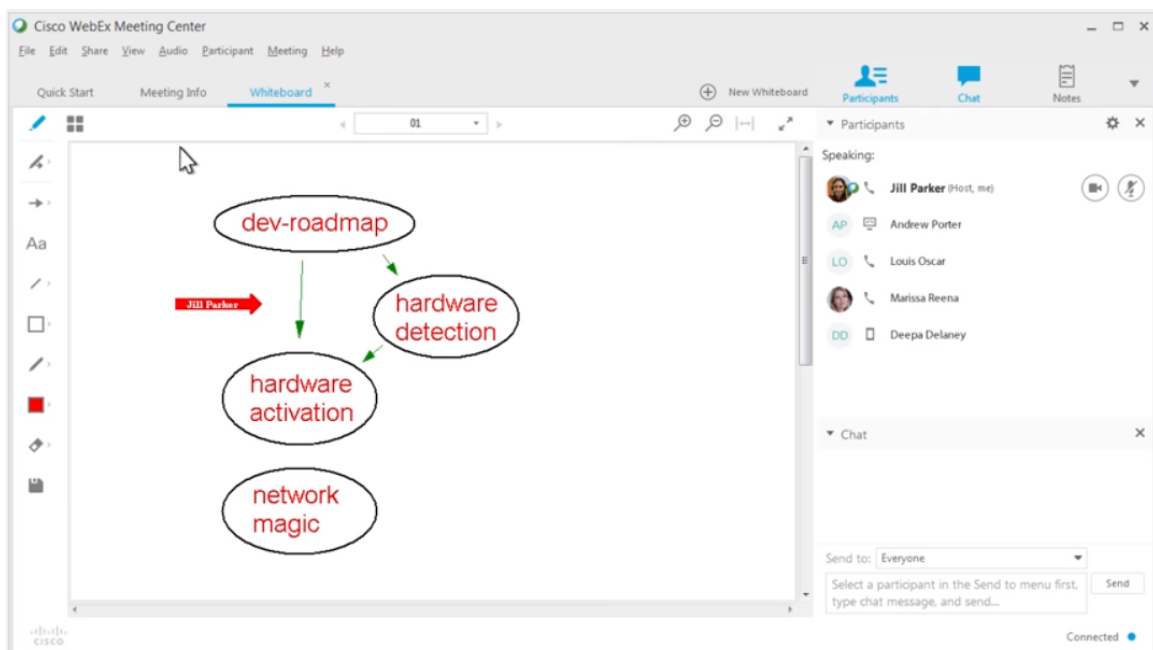


Figura 3.8: Interfaz de usuario de WebEx: a la izquierda se observa la pizarra colaborativa y a la derecha el panel de gestión de usuarios.

Compartir archivos

A través de Cisco WebEx, se pueden enviar archivos de forma segura entre los asistentes. El presentador puede controlar cuántos asistentes tienen su ventana de transferencia de archivos abierta y si han descargado los archivos. Además el presentador, puede mostrar un archivo en pantalla y usar los controles integrados para interactuar con el documento, por ejemplo, si está

compartiendo una presentación de PowerPoint, puede utilizar los controles para desplazarse por *slides* o cambiar a vista de pantalla completa.

Compartir pantalla

Los asistentes tienen la opción de compartir toda su pantalla o parte de su pantalla. Una vez que el elemento se introduce en la conferencia, los asistentes ven todo dentro de su visor de contenido y pueden realizar cambios en el documento en tiempo real utilizando las herramientas de anotaciones que ofrece WebEx.

3.4.3. Administración y sesiones

Grabado de sesiones

Permite a los usuarios grabar, archivar y reproducir reuniones.

Administración de usuarios y salas

WebEx ofrece una libreta de direcciones personal en línea, en la que se puede agregar información sobre contactos y crear listas de distribución. Al programar o iniciar una reunión instantánea, se puede invitar rápidamente cualquier contacto, o seleccionar una lista en lugar de cada miembro individualmente.

Posee salas personales, es decir salas de conferencias virtuales donde se puede iniciar una reunión o enviar invitaciones a reuniones sin tener que utilizar el calendario. El presentador puede bloquear su sala personal y controlar quién se une a la reunión. Dentro de la sala puede utilizar toda la funcionalidad de WebEx, como compartir contenido y video, y grabar reuniones. Los invitados pueden encontrar fácilmente estas salas ya que el enlace web nunca cambia y puede ser agregado a favoritos.

Calendario

WebEx posee un centro de eventos que permite a los anfitriones planificar reuniones e invitar a los asistentes a través de correo electrónico, teléfono o mensajería instantánea.

Sesiones públicas

Los participantes no necesitan tener una cuenta WebEx pero, al ingresar a la plataforma deben identificarse con un nombre, correo electrónico y conocer el número de salón al que van a asistir.

3.4.4. Respeto de la Experiencia de Usuario

Curva de aprendizaje

Además de que la interfaz del usuario de WebEx tiene un aspecto moderno y diseñado para mejorar la facilidad de uso, se encuentran disponibles manuales y tutoriales de video realizados por Cisco, que otorgan a los usuarios el conocimiento necesario para comenzar a utilizar el sistema.

Interfaz de Usuario

La interfaz de usuario se divide en 3 áreas principales:

- La barra de menús, donde se encuentran todos los comandos para compartir contenidos, gestionar participantes y ajustar las opciones de audio y video entre otros. Estas acciones también se encuentran en otras áreas ya sea mediante iconos, botones o como opciones del menú contextual.
- Los paneles de la derecha permiten interactuar con los participantes. El primer panel muestra la lista de los mismos indicando si están desde un teléfono, computadora o dispositivo móvil. Haciendo clic en un participante se le puede otorgar el control del mouse y teclado, dar permiso para hacer anotaciones, enviarle un mensaje de chat, silenciarlo o incluso expulsarlo. Debajo de la lista se encuentra el panel de chat. Estos paneles se pueden colapsar o expandir y su orden puede alterarse desde el menú de configuración.
- En el área de contenido, que se encuentra en el centro, es donde se muestran los documentos, videos, aplicaciones, pizarras o escritorios compartidos. A la izquierda de esta área se encuentran las herramientas de anotación, que pueden ser escondidas en caso de no querer utilizarlas. Arriba del contenido, se muestran los controles para avanzar entre las páginas de los documentos, hacer *zoom* y un botón para pasar a pantalla completa. En el modo de pantalla completa, el contenido compartido ocupa toda la pantalla y aparece

una nueva barra de herramientas (Figura 3.9) en la parte superior con las opciones más frecuentes y la opción para salir este modo.

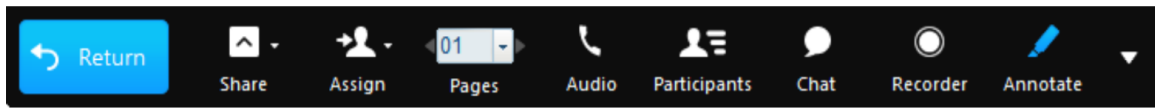


Figura 3.9: Barra de menú WebEx en modo pantalla completa.

Configuración inicial

Los primeros pasos con WebEx son muy simples, ya que sólo es necesario ingresar a la plataforma con las credenciales personales y una vez dentro es posible entrar a una sala personal y comenzar a enviar invitaciones, o bien programar una futura sesión. Con respecto a los participantes, WebEx permite agregar información sobre varios contactos simultáneamente en la libreta de direcciones personal, importando un archivo de valores separados por comas o tabulados (CSV)

3.4.5. Integración con otros sistemas

API

WebEx posee dos APIs públicas y abiertas para que los desarrolladores puedan usar y probar la plataforma, y una API para proveedores de servicios de teleconferencia.

- API de URL: se basa en HTTP (S) y proporciona un mecanismo de *hooks* sobre eventos originados en el navegador que permiten invocar a los servicios WebEx. La API de URL suele utilizarse en integraciones empresariales para programar, iniciar o unirse a reuniones e invitar a asistentes y presentadores.
- API XML: esta API es un conjunto completo de servicios, que soportan aspectos como la gestión de usuarios, funciones completas de administración y generación de reportes. Estos servicios responden a mensajes de solicitud XML que son enviados desde aplicaciones externas y devuelven una respuesta XML, ya sea indicando que se tomó una acción o devolviendo los datos solicitados (Figura 3.10). Este es el mecanismo preferido para la integración a nivel de aplicación, dado que la aplicación receptora es libre de analizar, utilizar y presentar los datos de cualquier forma deseada.

- API de proveedor de servicios de teleconferencia (TSP): una API que permite a los proveedores de servicios de teleconferencia integrar su servicio de audioconferencia con WebEx, incluyendo funciones como atención de llamadas entrantes, devolución de llamadas, enmudecer y poner modo *mute* a los participantes, conmutación de video, subconferencias e invitaciones por teléfono.

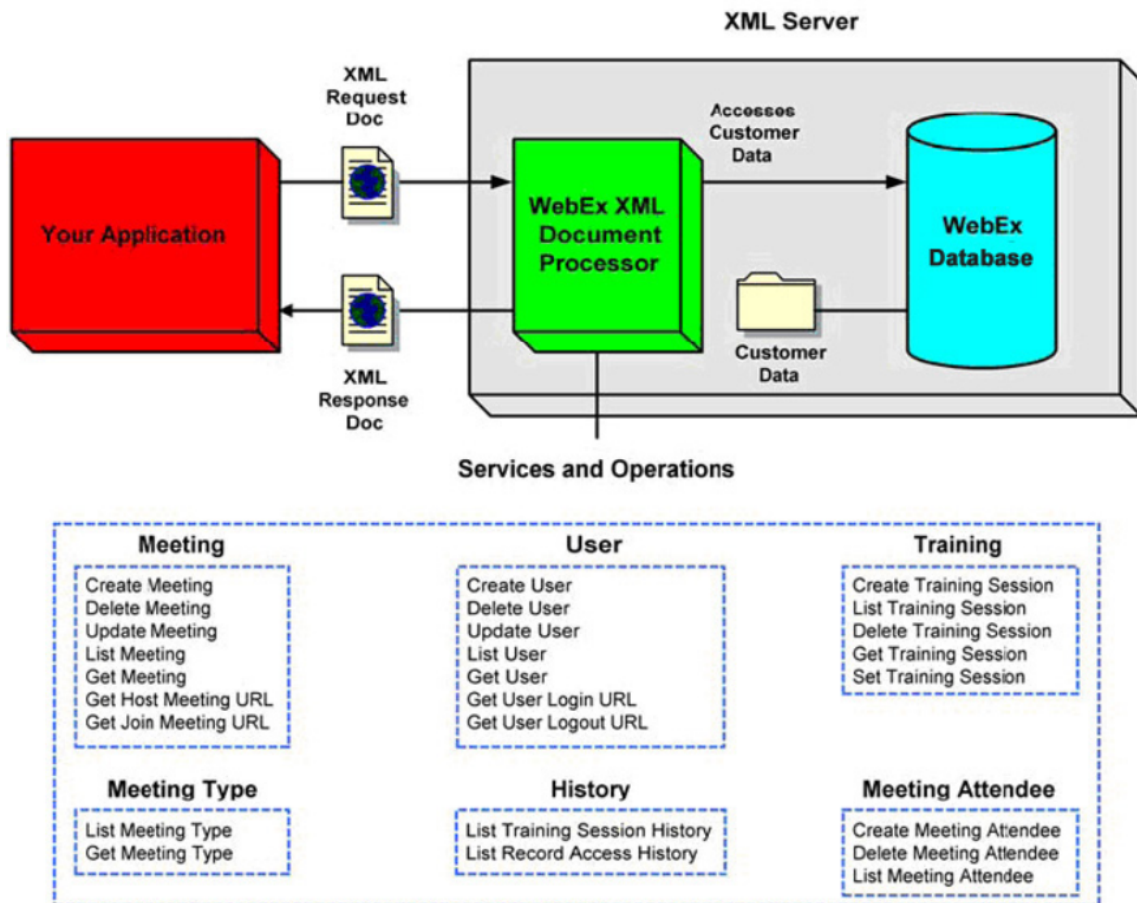


Figura 3.10: Arquitectura de la API XML de WebEx. Los servicios XML proveen funciones muy útiles como gestionar usuarios o programar reuniones.

Integración con otros productos

WebEx posee un conjunto de herramientas llamadas herramientas de productividad (*Productivity Tools*) que permiten integrar la funcionalidad de WebEx en un reducido grupo de softwares de terceros. Algunos de los softwares incluidos en estas herramientas son:

- IBM Lotus Notes: las herramientas permiten planificar, comenzar o unirse a una sesión

de WebEx utilizando el calendario de Lotus Notes.

- Microsoft Outlook: la integración provee un botón *Meet Now* que permite iniciar una sesión inmediata o abrir una sala personal. También provee botones para planificar o unirse a sesiones, ver la lista de sesiones planificadas y actualizar los datos personales.
- WebEx también posee integraciones con Microsoft Office y clientes de mensajería instantánea como Yahoo Messenger y Skype.

3.4.6. Otras características

Multiplés Idiomas

Además del inglés, Cisco WebEx puede ser usado en chino, francés, alemán, italiano, japonés, coreano, portugués, español, ruso, danés, sueco y turco.

3.5. Conclusiones

Luego de analizar en detalle las herramientas, se considera que tanto AOM como BBB son herramientas de software muy completas y adaptables a casi cualquier tipo de interacción entre varios usuarios. No obstante, se concluye que AOM, por su facilidad de instalación y la gran comunidad Apache que respalda el proyecto, es una mejor opción que BBB.

A costa de su flexibilidad, estas herramientas requieren de un cierto esfuerzo de mantenimiento, no sólo con respecto al monitoreo de errores o eventuales caídas del servicio, sino también en aspectos relacionados con la seguridad y el respaldo de datos.

Respecto de la herramienta de Cisco, en comparación con las otras dos, se considera más completa, más moderna y es la más fácil a la hora de comenzar a utilizar.

A su vez, el modelo de distribución como un servicio pago en la nube, permite a los usuarios centrarse en el uso de la herramienta evitando lidiar con aspectos como la instalación de nuevas actualizaciones o la seguridad y el *backup* de los datos. Sin embargo, este modelo resulta un limitante en dos aspectos principales:

- Por un lado el aspecto económico. Su elevado costo, deja fuera del alcance a instituciones educativas u organizaciones pequeñas que quieran probar el producto pero que no disponen de una reserva de dinero para gastos en tecnología. Sin embargo esto no impide que

la herramienta sea adoptada por empresas grandes y medianas, quienes prefieren optar por una solución lista para usar, y es además, ofrecida por una marca con trayectoria como Cisco, en lugar de montar una solución *open source*.

- Por otro lado, al utilizarse como servicio, no es posible disponer del código fuente de la herramienta, lo cual subordina a los usuarios a utilizar la funcionalidad que el proveedor del servicio quiera ofrecer en un determinado momento.

Al mismo tiempo, ninguna de las herramientas está adaptada especialmente para un modo de interacción docente/alumnos y, excepto por la opción para hacer encuestas, carecen de modos de interacción que le permitan a los docentes obtener una retroalimentación de su exposición.

Surge entonces, la idea de crear InfoMeeting a partir de las características más útiles de las herramientas vistas, junto con la incorporación de nuevos modos de interacción enfocados en el rol del docente.

Capítulo 4

InfoMeeting: descripción de su diseño

4.1. Introducción

En este capítulo se centra en la descripción del diseño de la herramienta InfoMeeting. Se empieza por presentar los motivos que dieron lugar a la propuesta de esta herramienta y luego se comienza a describir detalladamente la interfaz de usuario junto con sus componentes y características.

Asimismo, se especifican los diferentes roles de usuario y de qué manera interaccionan entre ellos. Se pone énfasis en los modos de uso de la herramienta y los diferentes componentes que la integran. A su vez, se detallan los flujos de interacción de las funcionalidades más relevantes que posee InfoMeeting.

Por último, se mencionan también, aquellas funcionalidades que serían deseables que la herramienta contemplara, pero que debido al alcance de este trabajo, no han podido ser desarrolladas y quedan abiertas como líneas futuras de trabajo.

4.2. Motivación

Las herramientas vistas en el capítulo 3, son en general completas en relación a las funcionalidades que ofrecen. No obstante, se considera que se pueden mejorar en algunos aspectos. En particular, se detectaron tres ejes de mejoras que se vinculan con: disponibilidad de acceso para dispositivos móviles, extensibilidad y usabilidad.

En cuanto a la disponibilidad de acceso para dispositivos móviles, en general hoy en día todos los software web se diseñan para que sean *responsive*. Sin embargo, hace unos años era normal encontrar sitios web contruidos en Flash, o alguna tecnología similar, que no podían ser ejecutados en el navegador de un sistema operativo como Android o iOS. De las herramientas analizadas previamente, solo WebEx cuenta con soporte para dispositivos móviles. Dicho esto y, dado el alto número de usuarios que acceden a la web mediante dispositivos móviles, se considera que cualquier plataforma web moderna debería poder ser accedida desde dichos dispositivos, siendo tal funcionalidad ya no una característica deseable, sino esperada.

Respecto de las tecnologías subyacentes utilizadas en los software analizados responden a características de generaciones previas limitando las posibilidades de desarrollo de hoy en día (éstas se mencionan en detalle en el capítulo 5). Este hecho limita la extensibilidad de las mismas a un reducido nicho de desarrolladores. Actualmente, hay nuevos protocolos aptos para el desarrollo de este tipo de herramientas, cuyas librerías están disponibles en lenguajes de programación modernos como JavaScript. Esto abre una puerta para que los programadores jóvenes puedan familiarizarse más rápidamente con el código del software, teniendo así la posibilidad de modificarlo o extenderlo.

Finalmente, respecto de la usabilidad, las herramientas vistas están pensadas para llevar a cabo sesiones desestructuradas. Sin embargo, en dinámicas de reuniones relacionadas con procesos de enseñar y aprender también es necesario contar con estrategias y formas de participación adicionales que atiendan a modos de trabajo tales como la exposición de temas y coordinación de las participaciones.

Teniendo en cuenta estos ejes analizados a los que no se atienden en las herramientas vistas, surge la idea de implementar InfoMeeting de manera tal de focalizar en estas necesidades.

4.3. Descripción general de InfoMeeting

InfoMeeting es una herramienta de comunicación sincrónica que permite a los usuarios participar de sesiones donde pueden comunicarse tanto por videoconferencia como por chat.

La dinámica de las sesiones fue pensada para que se adapte mejor a un tipo de interacción que involucre usuarios con los roles de alumnos y de tutores, siendo estos últimos quienes tienen la capacidad de controlar la sesión.

Para complementar la comunicación, InfoMeeting presenta algunas características adicionales como una herramienta para realizar consultas a los participantes y un editor de texto pensado para que los participantes de la sesión puedan escribir algoritmos de manera colaborativa.

A continuación se describen los aspectos más relevantes de InfoMeeting, detallando los componentes de su interfaz y la manera en que estos son utilizados por los diferentes roles de usuarios.

4.3.1. Roles de usuarios

Infomeeting cuenta con dos roles de usuarios (Figura 4.1): un usuario moderador y el usuario alumno.

El usuario moderador, tiene los privilegios necesarios para conducir la sesión, es decir que posee el primer plano del audio y video, puede escribir en el editor de código y puede limitar las acciones de los alumnos, según la dinámica de la sesión.

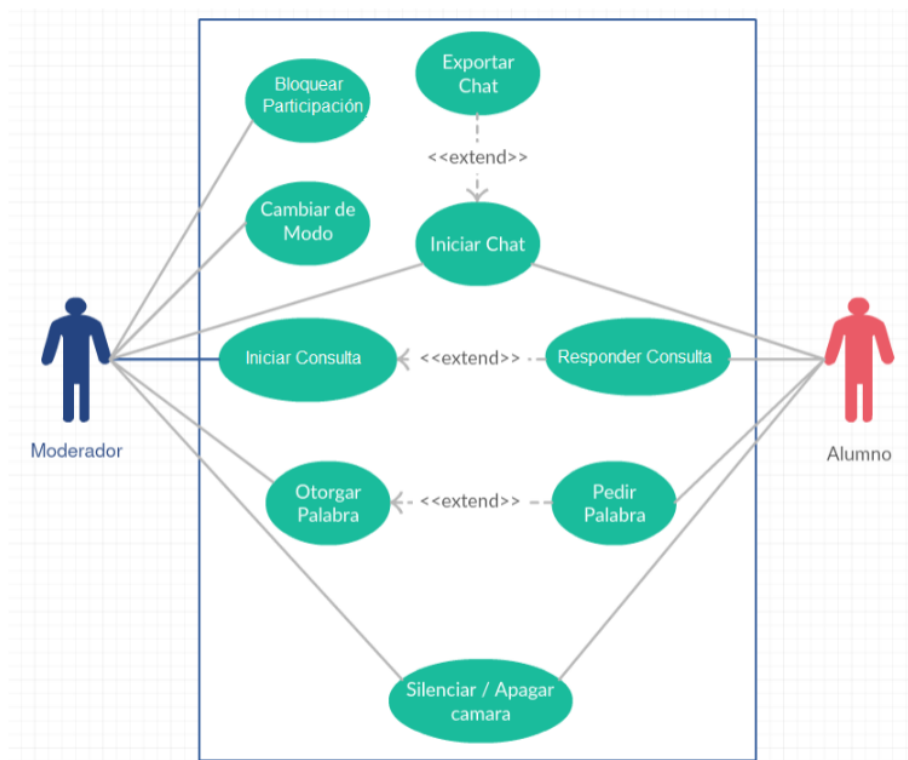


Figura 4.1: Diagrama de casos de uso de InfoMeeting. Se observan dos roles de usuario diferentes.

Los alumnos pueden ver y escuchar a sus moderadores. Además pueden ver en tiempo real como se actualiza lo que su moderador escribe en el editor de código. Los alumnos pueden utilizar el chat general e individual cuando lo deseen. El alumno, con la debida aprobacion del moderador, puede pasar a tener un rol más activo, pudiendo tener el control tanto del editor de código como del video y audio principal. Al mismo tiempo, InfoMeeting permite tener sesiones espontáneas sin moderador donde los alumnos pueden encontrarse para discutir o trabajar temas de interés. En ese caso, mientras no haya un docente presente, las reuniones adoptan una dinámica desestructurada.

4.3.2. Componentes de la interfaz

InfoMeeting cuenta con una pantalla principal la cual está compuesta por varios paneles o componentes (Figura 4.2): A. Video Principal, B. Editor de texto, C. Panel de usuarios, D1. Chat General, D2. Chat Individual y E. Video secundario.

También cuenta con una barra superior, la cual posee un menú con acciones. Cabe desta-

car que los datos y las opciones de cada sección varían de acuerdo al rol del usuario que se encuentra conectado al sistema en el momento.

La interfaz de usuario es similar a la de las herramientas analizadas en el capítulo 3. La ventaja más notoria con respecto a las herramientas vistas, es que la interfaz no está limitada a una pantalla de computadora de escritorio estándar, sino, que se adapta a cualquier tamaño de pantalla, incluyendo pantallas de dispositivos móviles. Al mismo tiempo habilita dinámicas de participación diferentes tales como las reuniones moderadas.

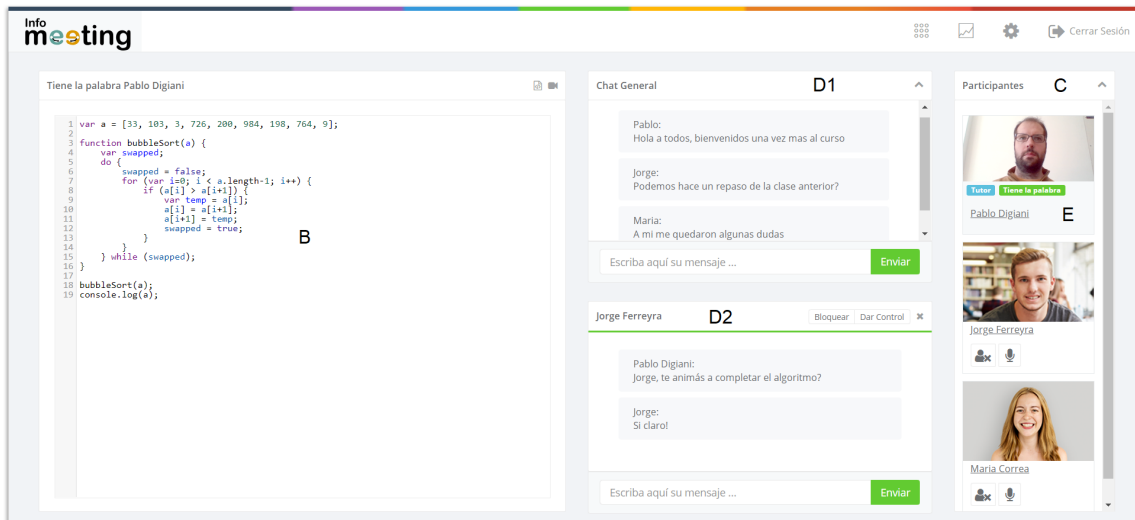


Figura 4.2: Interfaz de usuario de Infomeeting. La interfaz compone de varios paneles. El componente principal (izquierda) se encuentra en modo editor de código.

A continuación se describen en detalle los componentes mencionados.

A. Componente de video principal

Este es uno de los componentes principales de InfoMeeting y se encuentra ubicado en la parte superior izquierda de la pantalla. Este componente alberga la función de video (Figura 4.3), la cual permite ver y escuchar al usuario que tenga la palabra en el momento, ya sea este un docente o un alumno que haya solicitado la palabra.

Su tamaño es de aproximadamente la mitad del ancho de la pantalla, lo cual permite ver al moderador con suficiente detalle como para observar su lenguaje corporal, y a su vez, no interferir en la visualización del resto de los componentes de la interfaz.



Figura 4.3: El componente de video principal

B. Componente editor de código

Este componente (Figura 4.4) posee un área para editar texto de manera colaborativa. En particular se orienta a la edición de código y resalta en este caso las palabras claves del lenguaje JavaScript. Se consideró una funcionalidad interesante y específica para alumnos de carreras en Informática que abordan la creación de código como uno de los ejes de las carreras.



```
1 var a = ["a", "b", "c", "d", "f", "g", "h", "i", "j", "k", "l", "m", "n"];
2
3 // Recibe un array y el elemento a Buscar
4
5 function binarySearch(array, item){
6   var low = 0;
7   var high = array.length - 1;
8
9   while(low <= high) {
10    var middle = Math.floor((low + high)/2);
11    var guess = array[middle];
12    if(guess == item){
13
14      // Que valor deberiamos devolver?
15      return ??;
16    }
17    if(guess > item){
18      high = middle - 1;
19    } else {
20
21      // Como quedaria la variable low?
22      low = middle + 1;
23    }
24  }
25
26  // Que valor deberiamos devolver?
27  return ??;
28 }
29
30
31 binarySearch(a, "c");
```

Figura 4.4: Esta pantalla es un ejemplo de cómo el tutor de la sesión utiliza el editor de código para explicar un algoritmo

Está ubicado en el mismo lugar que el video principal. El cambio entre el video y el editor es efectuado por el usuario que posee el control de la sesión. Cuando el usuario decide pasar a otro modo, el componente cambia, y este cambio se ve reflejado en tiempo real para todos los usuarios conectados.

C. Panel de usuarios

En la parte superior derecha de la pantalla se encuentra el panel de usuarios (Figura 4.5), donde se muestra la lista de usuarios conectados en la sesión. Cuando la herramienta es operada por un docente, debajo de cada usuario se encuentran botones con opciones para bloquear la participación u otorgar el control de la sesión.

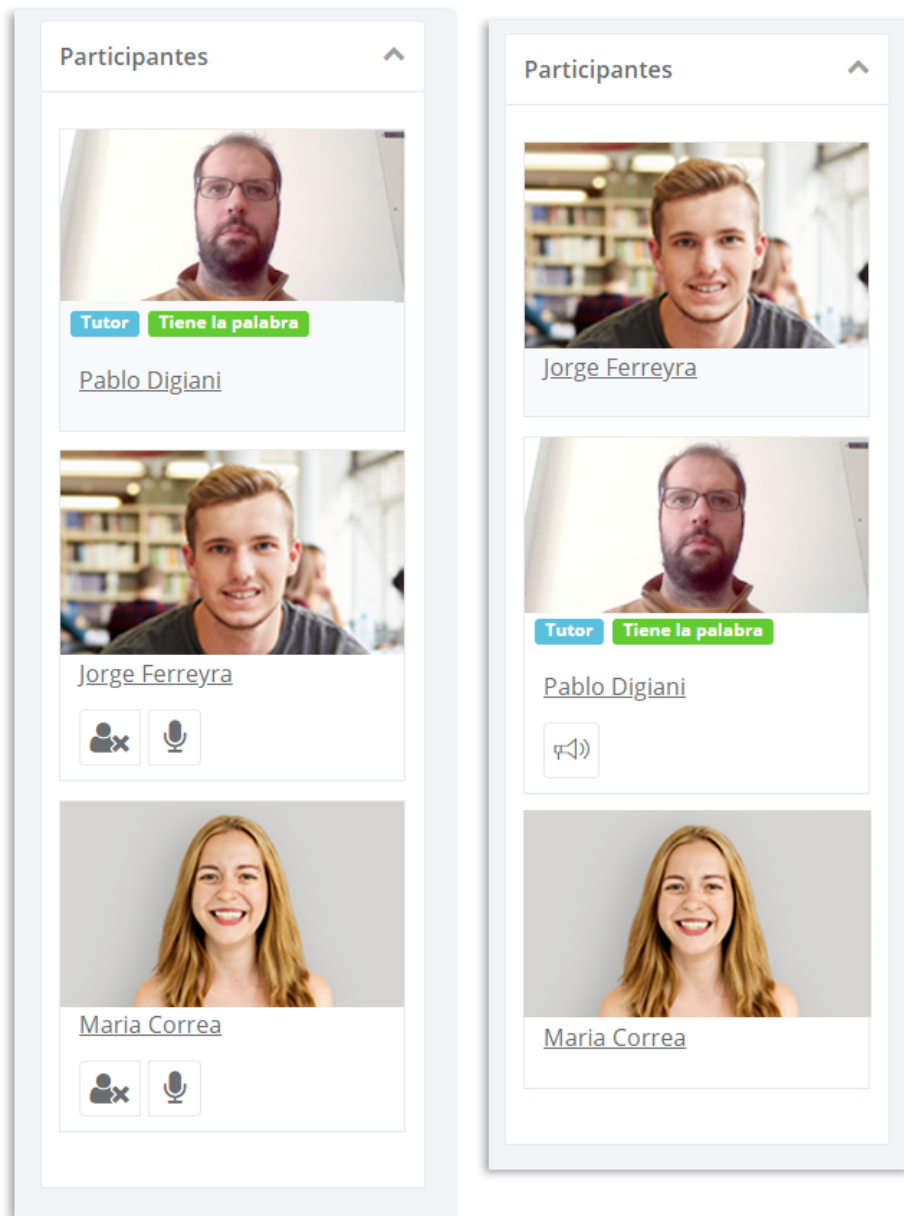


Figura 4.5: El panel de usuario visto por un docente moderador (izquierda) y por un alumno (derecha)

Cada usuario de la lista puede tener diferentes etiquetas informativas. Estas etiquetas son:

- Tiene el control: indica que el usuario posee el control actual del video principal y del editor de código.
- Pide la palabra: indica que el usuario quiere tomar el control. Esta etiqueta es solo visible por el moderador.

- Nuevo Mensaje: indica que hay un nuevo mensaje privado proveniente de ese usuario (Figura 4.6).
- Etiqueta del tutor: sirve para identificar al moderador de la sesión.

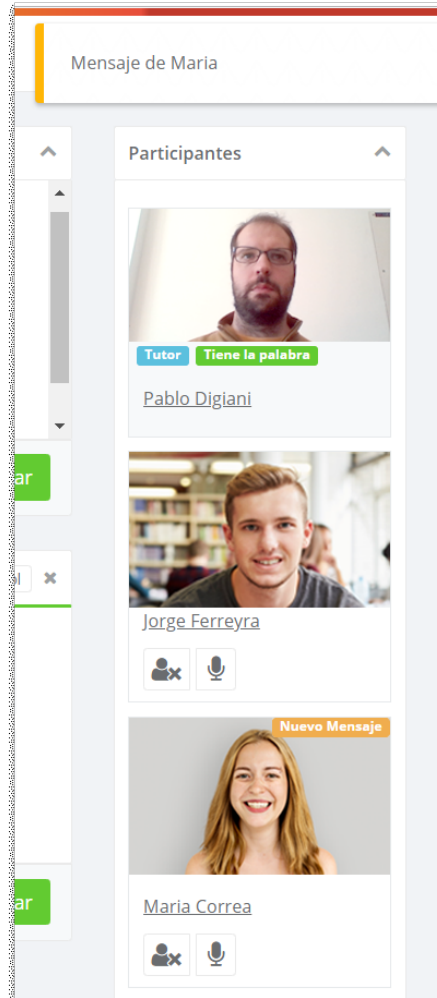


Figura 4.6: El moderador recibe notificaciones de un nuevo mensaje proveniente del usuario Maria Correa

D. Chat

El chat se encuentra ubicado entre el video principal y el panel de usuarios. Este componente se divide en dos paneles:

1. El panel de chat general, el cual se encuentra en la parte superior (Figura 4.7). Este componente está siempre visible. Los usuarios de la sesión pueden dejar en él sus mensajes,

los cuales serán vistos por todos los participantes.

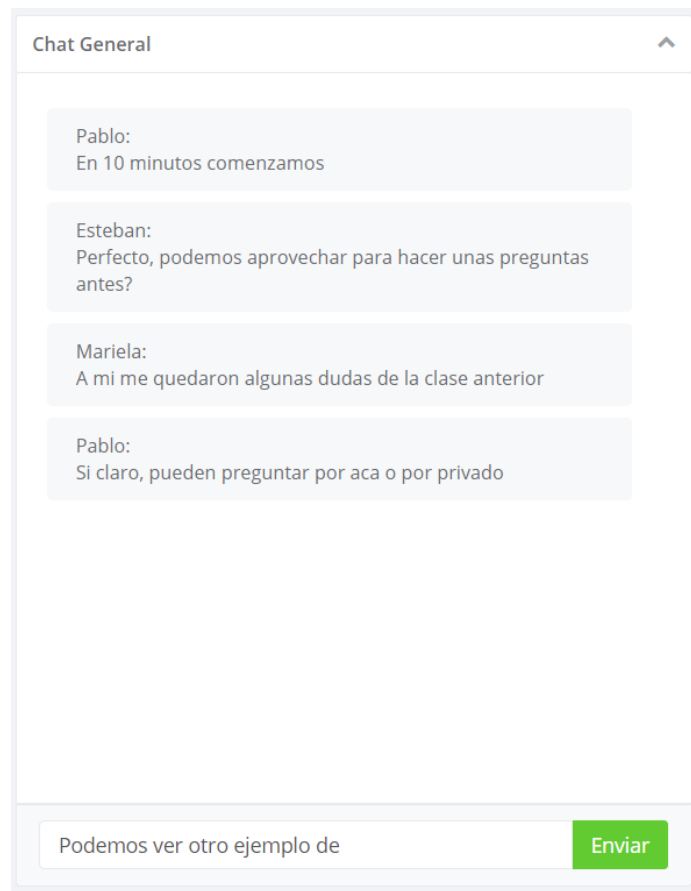


Figura 4.7: Panel de chat general

2. El panel de chat privado, el cual se muestra debajo del chat general (Figura 4.8). Este componente se hace visible al hacer clic en uno de los usuarios del panel. Permite que dos usuarios puedan comunicarse de manera privada. Al cerrar el chat privado, el componente desaparece y el chat general vuelve a ocupar todo el alto de la columna.

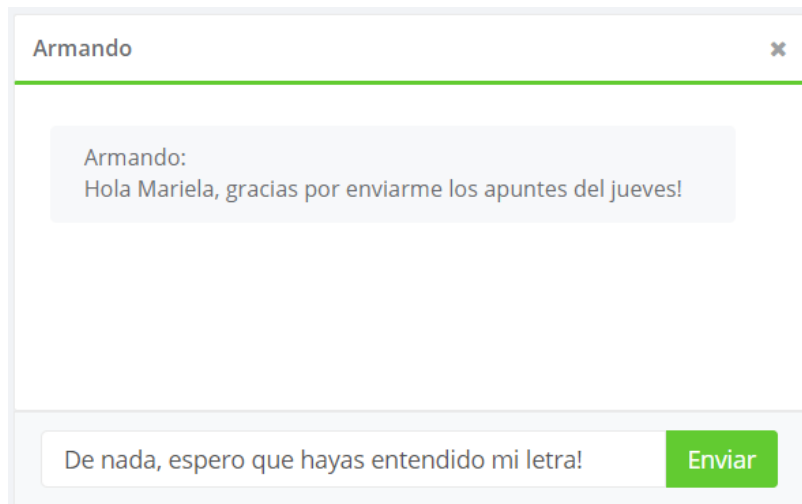


Figura 4.8: Chat privado entre dos alumnos

E. Video secundario

El video secundario se encuentra dentro del panel de usuarios, siendo el primer usuario de la lista. (Figura 4.9). Este componente, simplemente muestra el video del usuario conectado. Cuando el usuario conectado tiene la palabra, y el componente principal se encuentra en modo video, el video secundario desaparece, evitando mostrar dos veces las misma señal de video.

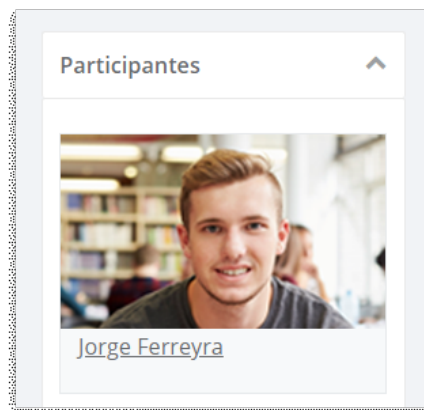


Figura 4.9: Componente de video secundario. Sólo se muestra cuando el usuario no tiene la palabra o cuando la herramienta está en modo editor de texto.

4.3.3. Interfaz para dispositivos móviles

La herramienta ha sido apropiadamente adaptada para usar en dispositivos móviles. La figura 4.10 muestra la vista principal de la herramienta estando en modo código. Con el fin de no sobrecargar de datos al usuario, los paneles de chat y de participantes vienen por defecto colapsados.



Figura 4.10: InfoMeeting vista desde un dispositivo móvil.

Por otro lado, como se observa en la figura 4.11, todas las interacciones se realizan a través del menú, permitiendo acceder cómodamente a las funcionalidades de la aplicación.

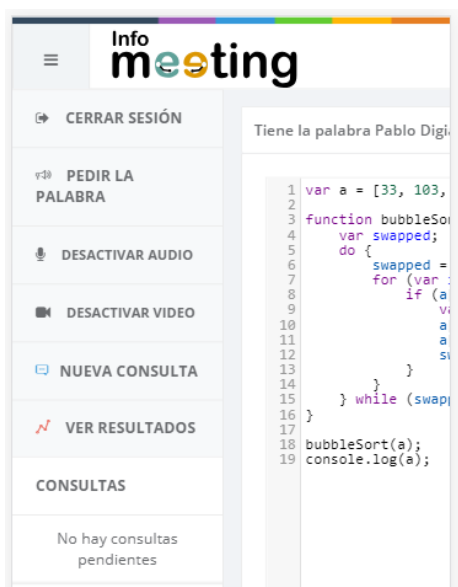


Figura 4.11: Menú de la aplicación en versión móvil.

4.3.4. Funcionalidades principales implementadas

Pedir la palabra

Una de las características más importantes de InfoMeeting es el concepto de pedir la palabra. Está inspirado en una dinámica de trabajo donde el profesor conduce la clase y los alumnos levantan la mano cuando quieren preguntar o responder algo, aunque podría adecuarse a modelo de participación entre alumnos o grupos en general. La implementación de este concepto se lleva a cabo de la siguiente manera.

Al iniciar la sesión, los usuarios tienen un rol, el cual es definido desde el EVEA desde donde se integra InfoMeeting. Cabe destacar que el rol de un usuario puede variar según el curso, es decir que puede ser un alumno en un curso de Matemática y, a la vez, un tutor en un curso de Programación. El primer participante tutor que se una a la sesión obtiene el rol de moderador de la misma, y por lo tanto, el control del video principal.

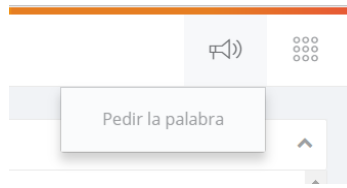


Figura 4.12: El usuario puede pedir el control (o devolverlo en caso de que ya lo tuviera) haciendo clic en el icono del megáfono en el menú superior.

Cuando un alumno desea intervenir en la clase, dispone de una opción en el menú superior, la cual es representada mediante un icono en forma de megáfono (Figura 4.12), desde la cual puede solicitar temporalmente el control de la sesión. Al hacer clic en la opción de pedir la palabra, se envía una petición al usuario moderador de la sesión y se muestra una confirmación (Figura 4.13) de que el pedido ha sido enviado.

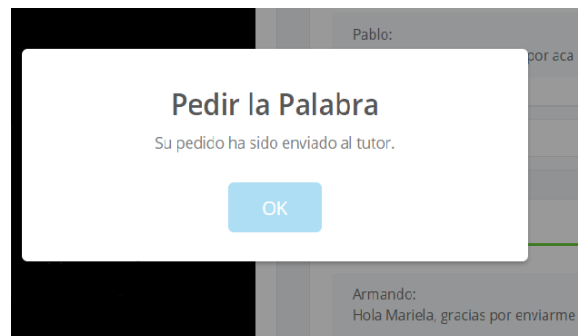


Figura 4.13: El usuario recibe una confirmación de que su pedido de palabra ha sido enviado.

Al recibir la petición, el tutor de la sesión, observa en el panel de participantes que tiene alumnos que quieren intervenir. Esto se muestra mediante la etiqueta *Pide la Palabra*, que aparece destacada en color verde junto al nombre del participante (Figura 4.14). El moderador entonces, aceptar el pedido haciendo clic en el botón dar control o simplemente puede ignorar el pedido hasta que lo considere necesario. En caso de proceder con el cambio de control, se mostrarán dos notificaciones: una para el moderador, indicando que la transferencia de control fue exitosa (Figura 4.15), y otra para el alumno, indicando que ahora posee el control de la sesión.

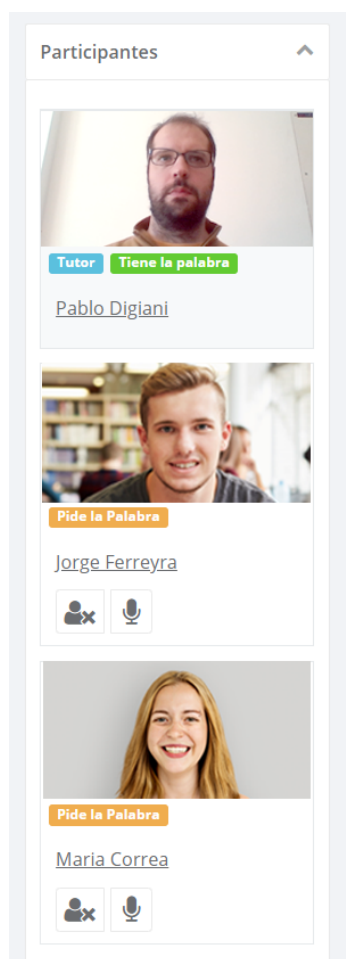


Figura 4.14: El moderador (Pablo) observa que los alumnos Jorge y Maria le piden la palabra.

Una vez que el alumno concluya su exposición, puede devolver la palabra desde el mismo menú donde inició el pedido. Si por algún motivo el moderador tuviera que intervenir, puede volver a tomar el control en cualquier momento, haciendo clic en la opción recuperar control, que se encuentra en el panel de usuarios junto al participante que tiene la palabra.

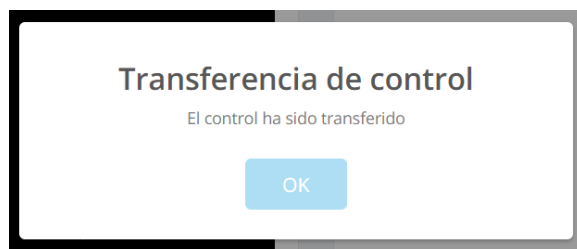


Figura 4.15: El moderador recibe una confirmación de que ha transferido el control de la sesión.

Realizar consultas a los participantes

InfoMeeting posee un módulo para realizar consultas entre los participantes de la sesión. Dicho módulo permite la creación de una consulta con una pregunta con respuesta tipo verdadero o falso. El moderador de la sesión puede iniciar una nueva consulta haciendo clic en el icono del gráfico de línea en el menú superior, que se encuentra al lado del icono para pedir la palabra (se puede agregar una figura). A continuación, se abrirá una ventana modal (Figura 4.16) donde el usuario podrá ingresar la pregunta correspondiente.

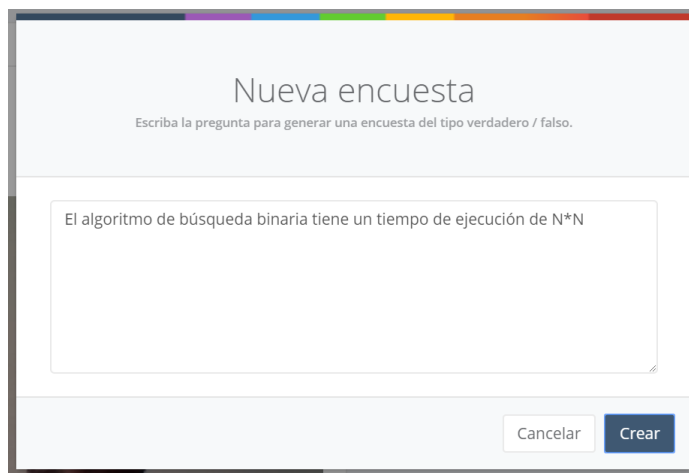


Figura 4.16: El moderador de la sesión crea una encuesta para ser respondida por los alumnos.

Una vez creada la consulta, los participantes de la sesión recibirán una notificación representada por un número sobre el icono de Consultas (Figura 4.17), indicando que hay una pregunta esperando ser respondida. Al hacer clic en el menú se desplegará una lista con las preguntas sin responder. El alumno puede hacer clic en una pregunta para ingresar su respuesta (Figura 4.18). Una vez que se responde, se muestra un mensaje para confirmar la respuesta elegida (Figura 4.19).

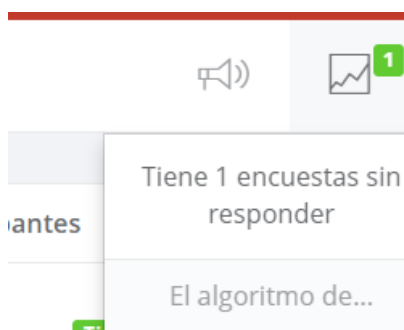


Figura 4.17: El alumno recibe una notificación de que posee encuestas sin responder.



Figura 4.18: Ventana modal de una encuesta. El alumno puede responder por verdadero o falso.

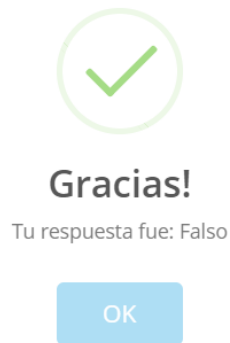


Figura 4.19: Confirmación de respuesta a la encuesta.

Bloquear participación

InfoMeeting pone a disposición del moderador la opción de bloquear a un participante de la sesión, en caso de que lo considerase necesario. Para efectuar tal acción, el moderador sólo debe elegir la opción Bloquear, que se encuentra en el panel de usuarios al lado del nombre del participante. Una vez efectuada la acción, el moderador recibirá una notificación de que se ha realizado la acción correctamente y el participante no podrá interactuar con la herramienta hasta ser revocada la acción. Esta funcionalidad está pensada para casos de superposición en el diálogo o situaciones excepcionales.

4.3.5. Persistencia de datos

Toda la actividad generada durante el transcurso de las sesiones de InfoMeeting es guardada en una base de datos relacional MySQL. Dichos datos también son expuestos en forma de servicio para que puedan ser recopilados y procesados por otras herramientas. En el siguiente capítulo se verá en detalle la estructura de la base de datos y la del servicio expuesto.

4.3.6. Funcionalidades que no se implementaron

Debido al limitado tiempo de desarrollo de este trabajo, se dejan como trabajo futuro algunas características analizadas como deseables de implementar. Entre ellas se mencionan la posibilidad de compartir pantalla, la subida de archivos, el grabado de sesiones y la visualización de resultados recopilados en las consultas.

Estas características están presentes en las herramientas analizadas en el capítulo 3, y se consideran de importancia dado que son funcionalidades que los usuarios asiduos de este tipo de herramientas esperan encontrar.

4.4. Conclusión

Se ha descrito la herramienta InfoMeeting, la cual presenta dos modos de interacción novedosos, como el concepto de pedir la palabra y el editor de texto colaborativo, siendo éste último especialmente útil para integrarse en el marco de procesos educativos de la Facultad de Informática.

Se ha detallado la interfaz de usuario de la herramienta, recorriendo todos sus componentes y viendo cómo estos funcionan y de qué manera se ven afectados según el rol del usuario activo.

Se mencionó también que las tecnologías utilizadas en la implementación de InfoMeeting son modernas en comparación a las otras herramientas analizadas. En el siguiente capítulo se profundizarán aspectos tecnológicos, explorando las librerías y protocolos que motivaron la realización de la herramienta.

Capítulo 5

Aspectos Tecnológicos de Infomeeting

5.1. Introducción

Este capítulo comienza analizando brevemente las tecnologías detrás de las herramientas vistas en el capítulo 3 y mencionando otras alternativas utilizadas para la implementación de aplicaciones de comunicación en tiempo real.

Luego se presenta en detalle la arquitectura de InfoMeeting, analizando las distintas capas que la componen y las tecnologías utilizadas. También se muestra la estructura de carpetas que utiliza Infomeeting y sus principales archivos y módulos.

Por otro lado, se describe el modelo de la base de datos empleada, que permite la persistencia de la información de las sesiones para su posterior análisis.

Finalmente se incluyen algunos detalles de la integración con el EVEA IDEAS y la manera en que InfoMeeting puede comunicarse con otros sistemas.

5.2. Análisis de las herramientas y tecnologías subyacentes para la implementación de InfoMeeting

En el capítulo 3 se realizó un análisis de diferentes herramientas para la comunicación sincrónica y se retoman aquí las tecnologías en las que se basa su implementación. De esta manera se constituyen las bases para la selección de aquellas tecnologías que se involucraron en InfoMeeting. Se descubrió que Apache Open Meetings está basado en OpenLaszlo, una plataforma para el desarrollo de aplicaciones web enriquecidas, cuyo desarrollo ha sido discontinuado. A su vez, tanto Apache Open Meetings como Big Blue Button, utilizan un servidor de medios llamado RED 5, que es un software libre desarrollado en Java, encargado de hacer la transmisión (*streaming*) de audio y video. Si bien el proyecto RED 5 continúa activo, la comunidad que lo desarrolla es un tanto pequeña en comparación con otros proyectos, y posee muy pocos desarrolladores activos.

Por otro lado, tanto Apache Open Meetings como Big Blue Button implementan su *frontend* utilizando una aplicación Flash, una tecnología conocida por tener problemas de acceso con dispositivos móviles ¹.

Con respecto a WebEx, no es posible hacer un análisis adecuado, dado que su modelo de negocios como servicio en la nube no permitió observar las tecnologías subyacentes.

A partir de este escenario, desde el punto de vista tecnológico, se observa que las herramientas vistas se encuentran implementadas con tecnologías que han dejado de ser estándares en la actualidad. Por tal motivo, se realizó una búsqueda de las tecnologías más modernas que se utilizan para el desarrollo de aplicaciones de similares características a las vistas.

Como resultado, se encontró que actualmente se tiende a utilizar WebSockets, una tecnología que hace posible una comunicación abierta entre el navegador de un usuario y un servidor, permitiendo enviar y recibir mensajes sin necesidad de iniciar una petición previa. También se encontró la tecnología WebRTC, un complemento apropiado para WebSockets, que permite la transmisión de audio y video de alta calidad.

El descubrimiento de estas tecnologías y el deseo de conocer más acerca de su funcionamiento y sus posibilidades, fueron de vital importancia a la hora de realizar el proyecto InfoMeeting, ya que han resultado un antecedente de importancia para este trabajo.

¹<https://www.apple.com/hotnews/thoughts-on-flash>

5.3. La arquitectura de Infomeeting

La arquitectura de Infomeeting puede observarse en la Figura 5.1. A simple vista se puede observar que se trata de una aplicación distribuida que consta de tres capas principales: una capa de persistencia, una capa de negocios o de lógica y una capa de presentación o front-end. La aplicación fue desarrollada íntegramente en el lenguaje de programación JavaScript. A continuación se describen las tecnologías elegidas para implementar cada capa.

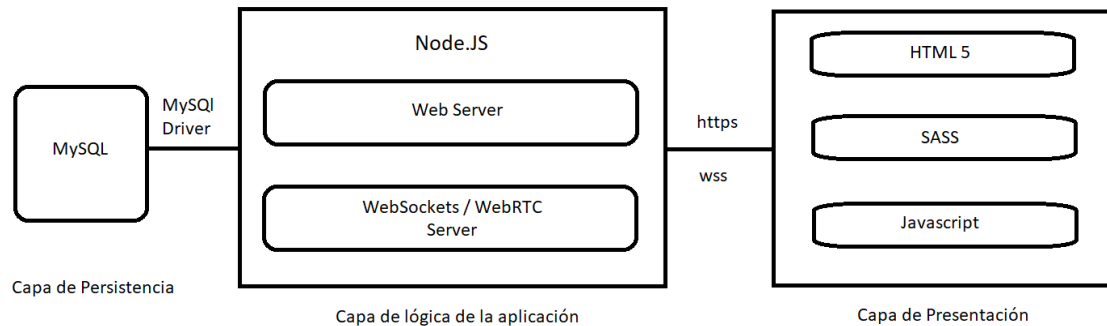


Figura 5.1: Arquitectura de la plataforma InfoMeeting.

5.3.1. Capa de lógica

La capa de lógica de la aplicación es la encargada de comunicarse con el cliente y con la base de datos. Esta implementada utilizando Node.js y Express.js, y permite la comunicación con el navegador del cliente a través de las tecnologías de WebSockets y WebRTC.

Node.js

Node.js es un ambiente de ejecución JavaScript (*runtime environment*) creado en el motor de JavaScript V8 de Chrome. Node utiliza un modelo de entrada / salida no bloqueante, el cual se encuentra controlado por un bucle o ciclo de eventos, lo que lo convierte en un ambiente liviano y eficiente.

Se entiende por entrada / salida no bloqueante el hecho de poder realizar llamadas al sistema (*system calls*) como por ejemplo realizar una escritura en disco, permitiendo que el procesamiento continúe antes de que haya terminado de realizarse la llamada.

En lugar de manejar procesos o hilos (*threads*) por cada conexión, Node usa un ciclo de

eventos (Figura 5.2) ejecutado en un único thread en el que registran *callbacks*, lo cual hace que sea escalable ya que sólo se asigna un pequeño tamaño de memoria para cada conexión. Una desventaja de este enfoque de un único subproceso es que Node no permite escalar verticalmente al aumentar el número de núcleos de CPU de la máquina en la que se está ejecutando.



Figura 5.2: Ciclo de eventos utilizado por Node.js.

A diferencia de otros servidores basados en eventos, el bucle de eventos de Node no necesita ser llamado explícitamente. Simplemente se definen los *callbacks*, y el servidor ingresa automáticamente el bucle de eventos al final de la definición del *callback*. Node sale del bucle de eventos cuando no hay más *callbacks* que realizar.

Estas características de su diseño, están apuntadas a optimizar la productividad y escalabilidad en aplicaciones web que cuenten con muchas operaciones de entrada salida, como también aplicaciones web en tiempo real, como herramientas de comunicación o juegos.

Otro aspecto destacable de Node es la forma en que gestiona los paquetes a partir del sistema NPM (*Node Package Manager*). Este sistema consiste en una herramienta de línea de comandos y una base de datos de paquetes públicos y privados llamada *npm registry*. Dichos paquetes son definidos en un archivo llamado *package.json* y pueden ser descargados simplemente con ejecutar el comando *npm install*. Desde su lanzamiento inicial a principios de 2010, el registro de NPM ha logrado acumular más de 477.000 paquetes ². Entre todos esos paquetes, podemos destacar al *framework* Express.js, el cual se describe a continuación.

²Ojamaa, Andres; Duuna, Karl (2012). "Assessing the Security of Node.js Platform". IEEE Xplore. Retrieved 22 July 2016

Express.js

Express.js es un *framework* para Node.js que permite la creación de aplicaciones webs o APIs de manera sencilla y minimalista.

Una instancia de Express, es una variable que representa un servidor web y permite atender conexiones en un determinado puerto. Sobre dicha instancia, se pueden agregar fácilmente las características que se desean implementar. En particular, las dos características más interesantes que ofrece Express son: el ruteo y los *middlewares*.

El ruteo o enrutamiento, es la manera en que se define como una aplicación responde a una solicitud por parte de un cliente para un determinado mensaje HTTP (usualmente GET y POST) en un determinado *endpoint*. Como se muestra en la Figura 5.3, Express.js permite definir rutas sobre una instancia de un servidor web utilizando un simple método, cuyo primer parámetro es el segmento del *endpoint* o URI y el segundo, la función de *callback* a ejecutar. Esta función de *callback* es invocada cuando la aplicación recibe una solicitud para la ruta y el método especificados. En otras palabras, la aplicación escucha las solicitudes que coinciden con la ruta y el método, y cuando detecta una coincidencia, llama al *callback* correspondiente.

```
var express = require('express')
var app = express()

// responde con "hola mundo" cuando se realiza una petición GET a la pagina principal
app.get('/', function (req, res) {
  res.send('Hola Mundo')
})
```

Figura 5.3: Definición de rutas en Express.js

Los métodos de enrutamiento pueden tener más de una función de *callback* como argumento, utilizándose en ese caso, un tercer parámetro dentro de la función de *callback* (que por convención se suele llamar next), con la próxima función a ejecutar.

Por otra parte, las funciones de *middleware* son funciones que tienen acceso al objeto de solicitud (req), el objeto de respuesta (res) y la siguiente función de *middleware* en el ciclo de solicitud-respuesta de la aplicación (next). Usualmente estas funciones se utilizan para ejecutar un determinado código o para modificar los objetos req y res. La Figura 5.4 muestra la definición de un *middleware* cuyo comportamiento es el de dejar un log de la dirección ip de cada conexión recibida.

```
var app = express()

app.use(function (req, res, next) {
  console.log('IP:', req.connection.remoteAddress)
  next()
})
```

Figura 5.4: Definición de un Middleware en Express.js

Debido a sus características, sumadas a la simplicidad de uso y a la rápida configuración, se eligió a Express como el *framework* para complementar a Node.js en el desarrollo de InfoMeeting.

WebSockets

WebSockets es un protocolo de comunicaciones que proporciona canales de comunicación *full-duplex* a través de una única conexión TCP.

Cuenta con una especificación para HTML5, la cual define una API que permite a las aplicaciones web utilizar este protocolo para lograr comunicación bidireccional con un host remoto. Define un canal de comunicación *full-duplex* que opera con un único socket a través de la Web y reutiliza la misma conexión del cliente al servidor, y del servidor al cliente. WebSockets no reemplaza al protocolo HTTP sino que se complementa con éste, iniciando la conexión con los clientes mediante un encabezado especial llamado *HTTP Upgrade*. En la Figura 5.5 se puede observar cómo se inicia la conexión, la cual permanece abierta hasta que alguna de las dos partes la cierra.

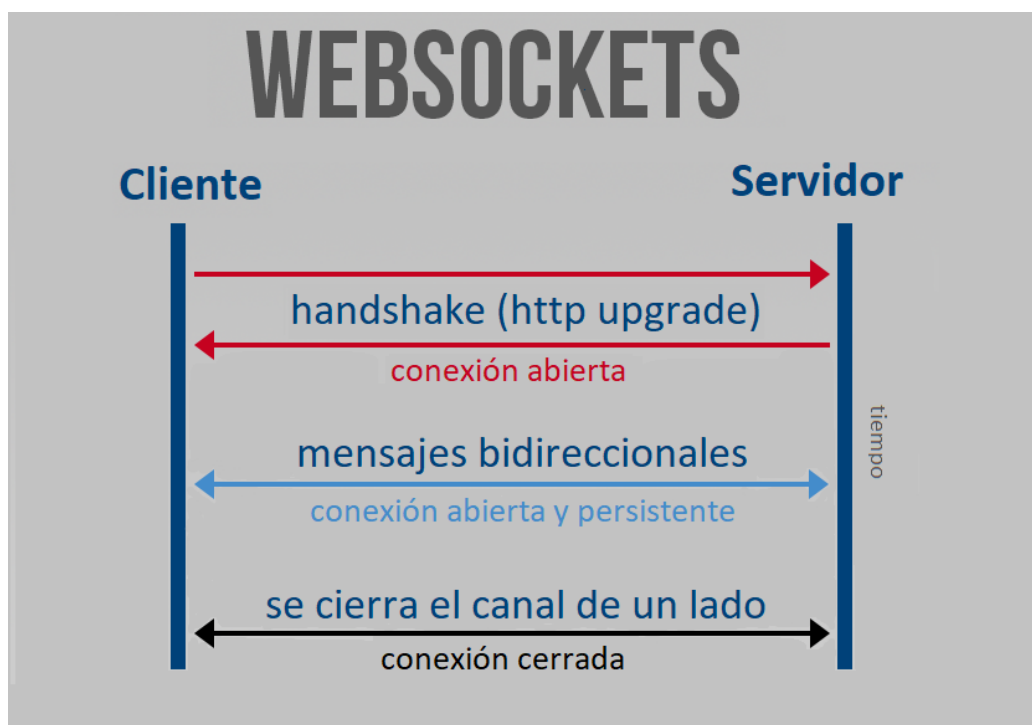


Figura 5.5: Representación visual de una conexión mediante WebSockets.

WebSockets reduce el tiempo de latencia porque una vez que se establece la conexión, el servidor puede enviar mensajes a medida que estén disponibles. Por ejemplo, WebSocket realiza una única petición a diferencia de otras técnicas como *polling*, donde se envía una solicitud a intervalos, independientemente de si hay mensajes disponibles. El servidor no necesita esperar una petición del cliente para enviarle datos. Del mismo modo, el cliente puede enviar mensajes al servidor en cualquier momento. Esta única petición reduce en gran medida la latencia.

Otra de las ventajas que ofrece WebSockets es su capacidad para atravesar *firewalls* y servidores *proxy*, un área problemática para muchas aplicaciones. Las aplicaciones tipo comet (un neologismo para describir un modelo donde un servidor puede enviar datos a un navegador sin que el navegador los solicite explícitamente) generalmente emplean *long polling* como una línea de defensa rudimentaria contra *firewalls* y servidores *proxy*. La técnica es efectiva, pero no es adecuada para aplicaciones que tienen requisitos de alto rendimiento.

WebSocket hace que la comunicación entre un cliente y un servidor, a través de la web, sea mucho más simple y mejore notablemente la interoperabilidad entre aplicaciones en tiempo real.

Las APIs y librerías disponibles son de usar y permiten lograr como resultado un código

legible y fácil de extender. La Figura 5.6 muestra la implementación (simplificada) del cliente InfoMeeting.

```
1  let conn = new WebSocket(getSocketUrl());
2
3  conn.onopen = (event) => console.log('Connected');
4  conn.onclose = (event) => console.log('Lost connection', event);
5  conn.onerror = (event) => console.log(event) ;
6
7  conn.onmessage = function (response) {
8
9      let data = JSON.parse(response.data);
10     switch (data.type){
11
12         // user interaction
13         case 'newuser': handleNewUser(data); break;
14         case 'changeActiveUser': handleChangeActiveUser(data); break;
15         case 'requestControl': handleRequestControl(data); break;
16         case 'changeMode': handleChangeMode(data); break;
17         case 'leave': handleLeave(data); break;
18
19         // chat
20         case 'chat': handleChat(data); break;
21         case 'singleChat': handleSingleChat(data); break;
22
23         // polls
24         case 'polls': handlePolls(data); break;
25         case 'pollAnswer': handlePollAnswer(data); break;
26         default: break;
27     }
28 };
```

Figura 5.6: Mensajes recibidos por el cliente de InfoMeeting.

De manera similar, se puede observar en la Figura 5.7 la implementación del lado del servidor. Es de notar que ambos códigos son muy similares: comienzan instanciando la librería y luego, del lado del cliente se conecta al servidor y queda en espera de nuevos mensajes, mientras que del lado del servidor, se crea la conexión (en el puerto 9090) y, al igual que como sucede en el cliente, se queda a la espera de nuevos mensajes.

```

1  let infoM = new InfoMeeting();
2  let wss = new WebSocketServer({port: 9090});
3  wss.on('connection', function(connection) {
4
5      connection.on('message', function(message) {
6
7          let data = JSON.parse(message);
8          switch(data.type) {
9
10             // interaction
11             case "editor": infoM.handleEditor(connection, data); break;
12             case "changeActiveUser": infoM.handleChangeActiveUser(connection, data); break;
13             case "requestControl": infoM.handleRequestControl(connection, data); break;
14             case "changeMode": infoM.handleChangeMode(connection, data); break;
15
16             // chat
17             case "chat": infoM.handleChat(connection, data); break;
18             case "individualChat": infoM.handleIndividualChat(connection, data); break;
19
20             // polls
21             case "createPoll": infoM.handleCreatePoll(connection, data); break;
22             case "pollAnswer": infoM.handlePollAnswer(connection, data); break;
23
24             default: InfoMeeting.SendTo(connection, {type: "error"}); break;
25         }
26     });
27
28     connection.on('close', function() {
29         infoM.closeConnection(connection);
30     });
31
32 });

```

Figura 5.7: Mensajes recibidos por el servidor de InfoMeeting.

WebRTC

Web Real-Time Communication (WebRTC) es una tecnología desarrollada por Google y lanzada como open source en mayo de 2011. Permite una comunicación entre navegadores, mediante la cual es posible el envío de audio y video sin la necesidad de usar *plugins* internos o externos.

De esta manera, WebRTC permite setear rápidamente conexiones peer to peer entre navegadores, algo que, en caso de querer construirse desde cero, necesitaría de una gran cantidad de *frameworks* y librerías para lidiar con problemas como la pérdida de datos, la caída de las conexiones o el NAT transversal. En cambio, con WebRTC, todas esas funcionalidades ya vienen implementadas dentro del navegador.

En comparación, otros métodos de comunicación entre navegadores utilizan un modelo cliente-servidor, en el cual todo el tráfico pasa por el servidor para que dos clientes se comuniquen. Con WebRTC, en cambio, el servidor sólo se usa para establecer las conexiones

iniciales entre los peers, permitiendo así, que el tráfico circule directamente entre ellos.

5.3.2. Capa de persistencia

La capa de persistencia de Infomeeting está basada en una base de datos MySQL.

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario, cuyo código está escrito en el lenguaje C. Puede ser usado tanto bajo la licencia GNU GPL, como con una licencia propietaria para aquellas empresas que quieran incorporarlo en sus productos comerciales.

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores web por su simplicidad.

Poco a poco los elementos de los que carecía MySQL fueron incorporados, tanto por desarrolladores internos como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

Dado que su facilidad de uso e instalación se mantiene vigente y que además, en la actualidad posee las características esperables de cualquier sistema de base de datos relacionales, se decidió usar MySQL como motor de base de datos.

Para la integración con InfoMeeting, se encontró en el registro de NPM, una librería llamada `mysql`, la cual presenta un driver escrito en JavaScript que permite la interacción entre aplicaciones Node y MySQL. La librería resulta sencilla de utilizar. En la Figura 5.8 se puede ver un ejemplo de cómo conectar a la base de datos y hacer una consulta de prueba.

```

1  let mysql = require('mysql');
2
3  let connection = mysql.createConnection({
4      host      : 'localhost',
5      user      : 'infomeeting',
6      password  : '#####',
7      database  : 'infomeeting_db'
8  });
9
10 connection.connect();
11
12 connection.query('SELECT 1 + 1 AS test', function (error, results, fields) {
13     if (error) throw error;
14     console.log('El resultado de la prueba es: ', results[0].solution);
15 });
16
17 connection.end();

```

Figura 5.8: Conexión a la base de datos con la librería mysql.

Cabe destacar que todas las operaciones que se realizan desde Node hacia MySQL son asincrónicas. Los usuarios que están acostumbrados a realizar operaciones con una base de datos de manera secuencial, tienen que adaptarse a la manera asincrónica, utilizando *callbacks*.

Una alternativa al problema de los *callbacks* es utilizar promesas (*promises*). Estos son objetos que actúan como un *proxy* para un resultado de una función que inicialmente no se conoce, usualmente porque aún se encuentra en ejecución. De esta manera el código final se simplifica notablemente. La Figura 5.9 muestra un ejemplo de un pseudocódigo con *callbacks* en la parte superior, y en la parte inferior el mismo código utilizando *promises*.

```

// codigo usando callbacks
connection.query( 'SELECT * FROM una_tabla', ( err, rows ) => {
  connection.query( 'SELECT * FROM otra_tabla', ( err, rows2 ) => {
    connection.close( err => {
      // ... hacer algo con los resultados
    })
  })
});

// codigo usando promises

database.query( 'SELECT * FROM una_tabla' )
  .then( rows => database.query( 'SELECT * FROM otra_tabla' ) )
  .then( rows => database.close() );

```

Figura 5.9: Comparación de una consulta hecha con *callbacks* (parte superior) y con *promises* (parte inferior).

5.3.3. Capa de presentación

Si bien las tecnologías de la capa de lógica parecen más complejas o difíciles de usar que las de la capa de presentación, la cantidad de opciones disponibles en el lado del *frontend* ha crecido notablemente en los últimos años. A continuación, se describen los componentes de esta capa, comenzando por el lado de JavaScript.

ES6, Babel y Webpack

Dada la gran variedad de librerías y tecnologías disponibles en JavaScript, se hace necesario hacer un repaso de la historia de JavaScript y definir algunos conceptos claves para poder tener una mejor comprensión del tema.

ECMAScript es un estándar para lenguajes de scripting. La implementación más conocida de este estándar, es el lenguaje JavaScript. Otras menos conocidas son los lenguajes JScript y Action Script.

Es importante mencionar que JavaScript surgió antes que ECMAScript. Fue desarrollado inicialmente con la intención de crear un lenguaje similar a Java para ser usado en la web por el navegador Netscape. Dado que Java era muy complicado, se creó un lenguaje más simple. Unos años después, Internet Explorer tomó el lenguaje existente y comenzó a hacer sus propias implementaciones (JScript).

Antes de la caída de Netscape, los autores decidieron crear el standard con la idea de guiar el desarrollo de JavaScript. ECMA tuvo varios lanzamientos desde entonces, siendo el último de ellos (sexta edición) en 2015, la cual incorporó nuevas características. Algunas de ellas son:

- Funciones arrow: son abreviaciones de funciones que se declaran utilizando el operador `=>`. A diferencia de las funciones normales, las funciones arrow comparten el alcance (scope) con el código que las rodea.
- Herencia de clases y soporte para métodos estáticos.
- Objetos literales mejorados: ES6 permite la inicialización de objetos definiendo una lista de propiedades entre llaves (`{}`)
- Módulos: ES6 se dispone de una manera para cargar y gestionar dependencias a través de las palabras claves `import` y `export`. Esto permite una reasignación fácil de código global a código modular y facilita la carga externa de código de manera asincrónica.
- Promises: como se vio en la sección de MySQL, ES6 es un lenguaje basado en *promises*, es decir que por cada operación asincrónica hay un promise para manejar el resultado de la ejecución de dicha operación.

Dado que al momento de realizar este trabajo, aún existen varios navegadores que no soportan todas las características de ES6, para poder utilizarlas fue necesario recurrir a una herramienta conocida como “transpilador” (*transpiler*) que permite convertir un código fuente en otro código fuente, en este caso ES5 a ES6.

La herramienta utilizada para hacer el “transpilado” del código fue Babel, un proyecto comunitario mantenido por un grupo de voluntarios.

Dado que del lado del navegador no es posible utilizar funciones como `include` or `require` para poder acceder al contenido de otros archivos, se hace necesario unificar todo el código JavaScript de la aplicación en un único archivo. La herramienta que se utilizó para esta tarea es Webpack, un analizador de dependencias capaz de construir un paquete o *bundle*. La manera de definir este proceso es mediante el uso de un loader, el cual es un script que procesa los archivos a medida que se los incluye. Existe un paquete llamado *babel-loader*, el cual permite a Webpack armar los bundles a partir del código generado por Babel.

JQuery

Para implementar el *frontend* de la aplicación se utilizó la librería jQuery para facilitar la manipulación de los elementos del DOM durante el renderizado de los componentes. Se eligió usar jQuery por su facilidad de uso, ya que su sintaxis permite reducir el número de líneas que se necesitan llevar a cabo en muchas tareas. En particular se destaca la facilidad para agregar eventos a los elementos de la aplicación. Además, al ser un producto maduro con cantidad de usuarios y varios años en el mercado, dispone de una gran variedad de tutoriales y de documentación

A pesar de haber elegido jQuery, se sostiene que desde un punto de vista arquitectural, existen mejores alternativas para trabajar con vistas y componentes. Por ejemplo, hay *frameworks* como React o Vue.js, que utilizan un DOM virtual y proveen componentes reactivos y combinables. Mientras que en el caso de jQuery, el estado de la aplicación se mantiene directamente en el DOM. Sin embargo, dada la elevada curva de aprendizaje de tales tecnologías, se decidió no utilizarlas para el desarrollo de InfoMeeting.

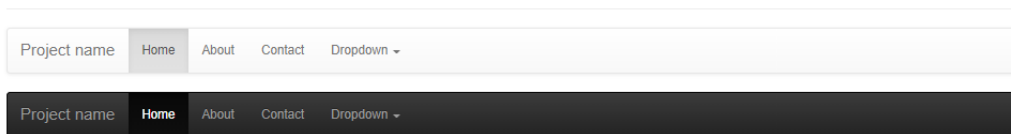
Bootstrap

Bootstrap es un *framework* de CSS, es decir es una librería de estilos genéricos usada para implementar diseños web, la cual aporta una serie de utilidades que pueden ser aprovechadas frecuentemente entre distintos diseños. Si bien los *frameworks* CSS pueden restringir la creatividad, ya que el desarrollador se tiene que amoldar a la estructura provista, las utilidades disponibles permiten ahorrar un tiempo considerable durante el desarrollo de un proyecto.

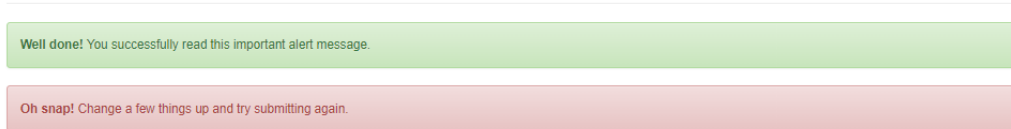
El nombre original del proyecto Bootstrap fue Twitter Blueprint, ya que fue creado por el equipo de desarrolladores de Twitter, con el propósito de promocionar la consistencia entre sus herramientas internas.

A diferencia de otros *frameworks* CSS, Bootstrap no sólo posee una base de clases para utilizar asegurando la compatibilidad entre navegadores, sino que además posee soporte para varios componentes de JavaScript, como paneles, ventanas modales, botones, carruseles y acordeones (Figura 5.10).

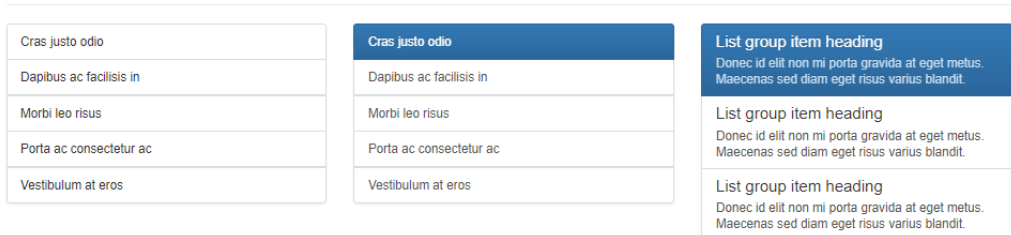
Navbars



Alerts



List groups



Panels

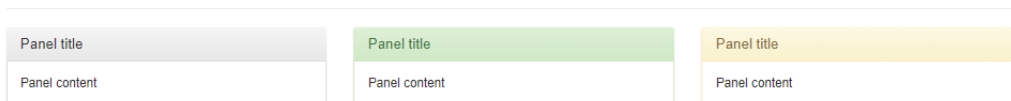


Figura 5.10: Algunos de los componentes que ofrece del *framework* Bootstrap.

Dadas las ventajas mencionadas, sumadas al respaldo de una documentación completa y una gran comunidad que lo soporta³, hicieron que Bootstrap sea la librería elegida para el desarrollo de InfoMeeting.

SASS

Uno de los inconvenientes que se producen al incrementar la complejidad de una aplicación web, es mantener las hojas de estilos CSS. Esto sucede porque CSS provee una funcionalidad muy simple y no está pensado para poder componer estilos pensando en la reusabilidad.

Una alternativa para escribir mejor código CSS es separar las definiciones en pequeños archivos e importarlos en un archivo principal. Sin embargo, esta técnica no resuelve problemas de repetición de código.

³<https://getbootstrap.com/docs/3.3/about>

Otro enfoque es dividir una clase en varios estilos, es decir, un estilo para los colores, otro para el espaciado, otro para las fuentes, etc. Si bien esta técnica favorece la reusabilidad, por otro lado reduce la mantenibilidad.

Es entonces donde surgen los preprocesadores de CSS, los cuales permiten escribir estilos agregando conceptos de lenguajes de programación, como variables, y *mixins*. Estas tecnologías si bien están presentes hace varios años, han cobrado popularidad en los últimos años, debido a la creciente complejidad de las aplicaciones web.

El preprocesador utilizado por InfoMeeting es SASS. Este lenguaje de *scripting* implementa las mencionadas características de los lenguajes de programación con una sintaxis fácil de entender, y permite así poder generar un código CSS mantenible y reusable.

5.4. La estructura de archivos de InfoMeeting

A continuación se describe la estructura de carpetas y archivos que componen el código de InfoMeeting. Las descripciones de los archivos más importantes se complementan con algunas figuras para una mejor comprensión de los mismos.

- Bin: esta carpeta contiene los archivos de inicialización de la herramienta.
 - Dev: contiene el script para iniciar la herramienta en modo desarrollo. Esto incluye las directivas para realizar la traducción del código utilizando Babel.
 - Prod: contiene el script para iniciar la herramienta en modo producción. En este modo, el código JavaScript (del lado del cliente) ya se encuentra generado y optimizado.
- Public: incluye el código relacionado con el *frontend* de la aplicación.
 - Assets: contiene fuentes, imágenes y otros archivos.
 - Fonts: contiene las diferentes fuentes utilizadas en InfoMeeting, entre ellas la fuente para los iconos font-awesome.
 - Images: contiene los logos, banners y algunas las imágenes de prueba.
 - Sass: contienen los archivos SASS utilizados durante el desarrollo.
 - Vendor: contiene varias librerías, en su mayoría dependencias del *framework* Bootstrap.
 - Styles: contienen los archivos CSS generados y listos para ser cargados.
 - Theme.js: contiene directivas para la inicialización de componentes JavaScript.

- Main.js: es el archivo principal de InfoMeeting, desde donde se conecta al servidor y se emiten y reciben todos los mensajes. La Figura 5.11 muestra un extracto del código.
- Scripts.js: contiene el código JavaScript unificado y optimizado, listo para ser cargado por el navegador.
- Modules:
 - Chat.js: contiene el código para renderizar el panel de chat general e individual.
 - Polls.js: contiene el código para renderizar los paneles relacionados con las encuestas.
 - UserPanel.js: contiene el código para renderizar el panel de usuarios.

```

125     $('#btn-request-control').on('click', () => {
126         if(!isAdmin() && !hasControl()){
127             if(appState.sesion.usuario_moderador){
128                 send({type: 'requestControl', user: user});
129                 swal({
130                     title: "Pedir la Palabra",
131                     text: "Su pedido ha sido enviado al tutor."
132                 });
133             }else{
134                 swal({
135                     title: "Pedir la Palabra",
136                     text: "Por favor espere a que un tutor se una a la sesión."
137                 });
138             }
139         }
140     });

```

Figura 5.11: El archivo main.js inicializa los eventos de la interfaz.

- Server: esta carpeta contiene todo el código de la capa de lógica de InfoMeeting
 - Index.js: es el archivo principal desde donde se instancia el servidor web, el servidor de WebSockets y se atienden las conexiones de los clientes.
 - Infomeeting.js: contiene el código para responder a los diferentes tipos de mensajes enviados por los clientes. La Figura 5.12 muestra las funciones definidas en la clase.
 - Encuesta.js: contiene el código relacionado con las encuestas.
 - Sesion.js: contiene el código relacionado con las sesiones.
 - Usuario.js: contiene el código relacionado con los usuarios.
- Config.js: contiene los parámetros básicos para iniciar la aplicación (Figura 5.13) como el puerto, los datos para conectar a la base de datos, etc.

- Database.js: contiene el código para manejar las conexiones a la base de datos.
- Package.json: archivo de configuración de NPM.
- Webpack.js: archivo de configuración de Webpack.

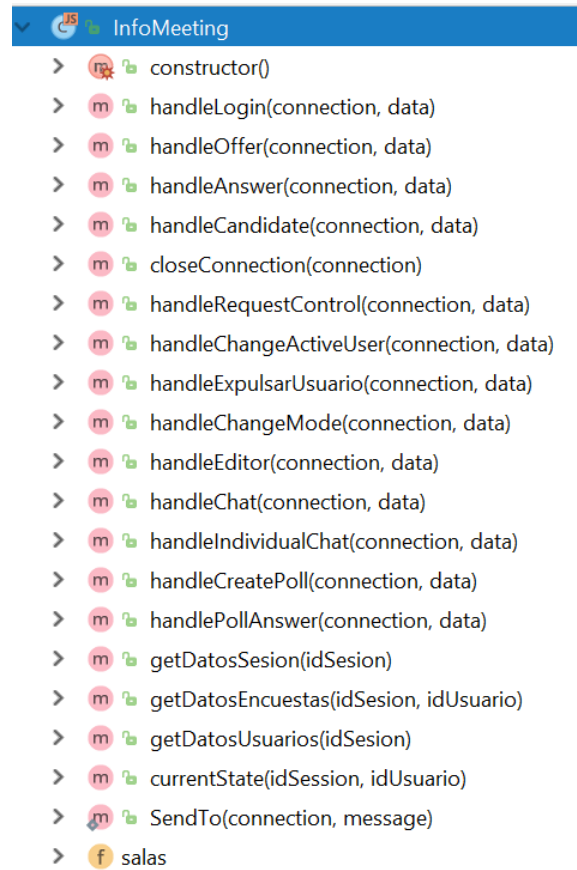


Figura 5.12: Estructura de la clase InfoMeeting

```

1  let config = {};
2
3  config.puerto = 'numero de puerto';
4  config.modo = 'poster o video';
5
6  config.db = {};
7  config.db.host = 'host de la base de datos';
8  config.db.usuario = 'usuario para acceder a de la base de datos';
9  config.db.password = 'password para acceder de la base de datos';
10 config.db.nombre = 'nombre de la base de datos';
11
12 config.ssl = {};
13 config.ssl.certificado = 'ruta del certificado';
14 config.ssl.llave = 'ruta de la llave privada del certificado';
15
16 module.exports = config;

```

Figura 5.13: El archivo config.js permite definir los parámetros básicos de InfoMeeting.

5.5. Integración con IDEAS

Dado que InfoMeeting no cuenta con una base de datos de usuarios propios, debe implementar un mecanismo de autenticación utilizando algún servicio externo. Además cuando se plantearon los objetivos de esta tesina, uno de ellos se orientó a la integración de esta herramienta en un entorno virtual de enseñanza y aprendizaje (EVEA IDEAS)

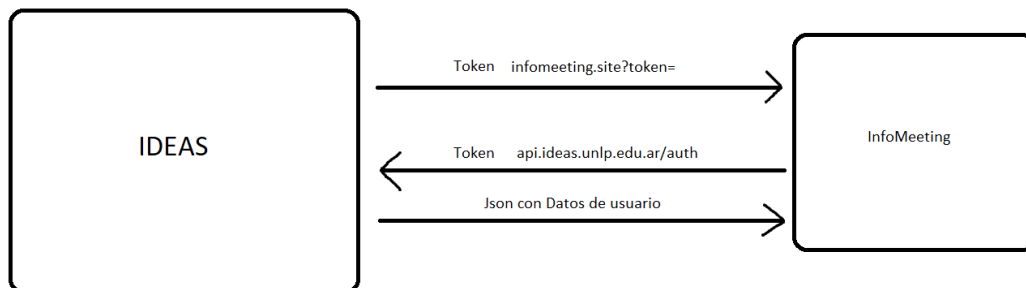


Figura 5.14: Integración entre IDEAS e InfoMeeting.

El mecanismo propuesto, para autenticar utilizando IDEAS, se resume en la Figura 5.14 y consiste en los siguientes pasos:

- Los usuarios se autentican normalmente en IDEAS con su nombre de usuario y contraseña. Cuando el usuario accede a un curso, el EVEA IDEAS genera un token, el cual permite identificar un usuario dentro de un curso.

- Este token es agregado a la URL de InfoMeeting en el botón o el enlace que permite a los alumnos acceder a InfoMeeting (ej: <https://infomeeting.site?token=nd73120bh531d>)
- Cuando el usuario accede a esa URL, InfoMeeting obtiene el valor del token y realiza una petición a un servicio de validación de usuarios expuesto por IDEAS.
- El servicio debería hacer una búsqueda por usuario en base al token recibido y retornar los datos correspondientes. Ej:

```
{
  "_id": "5a0d74c83b4d1af7680f8c1e",
  "nombreUsuario": "juangonzalez",
  "nombreCompleto": "Juan Gonzalez",
  "curso": "redesneuronales",
  "rol": "tutor",
  "avatar": "http://placeholder.it/32x32"
}
```

Figura 5.15: Token recibido por InfoMeeting

Con la respuesta obtenida, InfoMeeting ya tiene la información necesaria para mostrar los datos del usuario recientemente ingresado y definir el alcance de su funcionalidad.

5.6. Persistencia de datos

La capa de persistencia tiene como objetivo guardar toda la información generada durante el desarrollo de una sesión en InfoMeeting. Como se mencionó anteriormente, se utiliza una base de datos MySQL para llevar a cabo la persistencia de los datos.

El modelado de la base de datos se basa principalmente en una tabla sesión, la cual se relaciona con casi todo el resto de las tablas. Los registros de esta tabla se generan cuando un participante ingresa por primera vez a la herramienta en un determinado curso. Cuando el último usuario de una sesión cierra su navegador, InfoMeeting actualiza la columna `fecha_fin` en la tabla sesiones, y, de esa manera, la sesión se da por finalizada.

La Figura 5.16 muestra el modelo relacional de la base de datos.

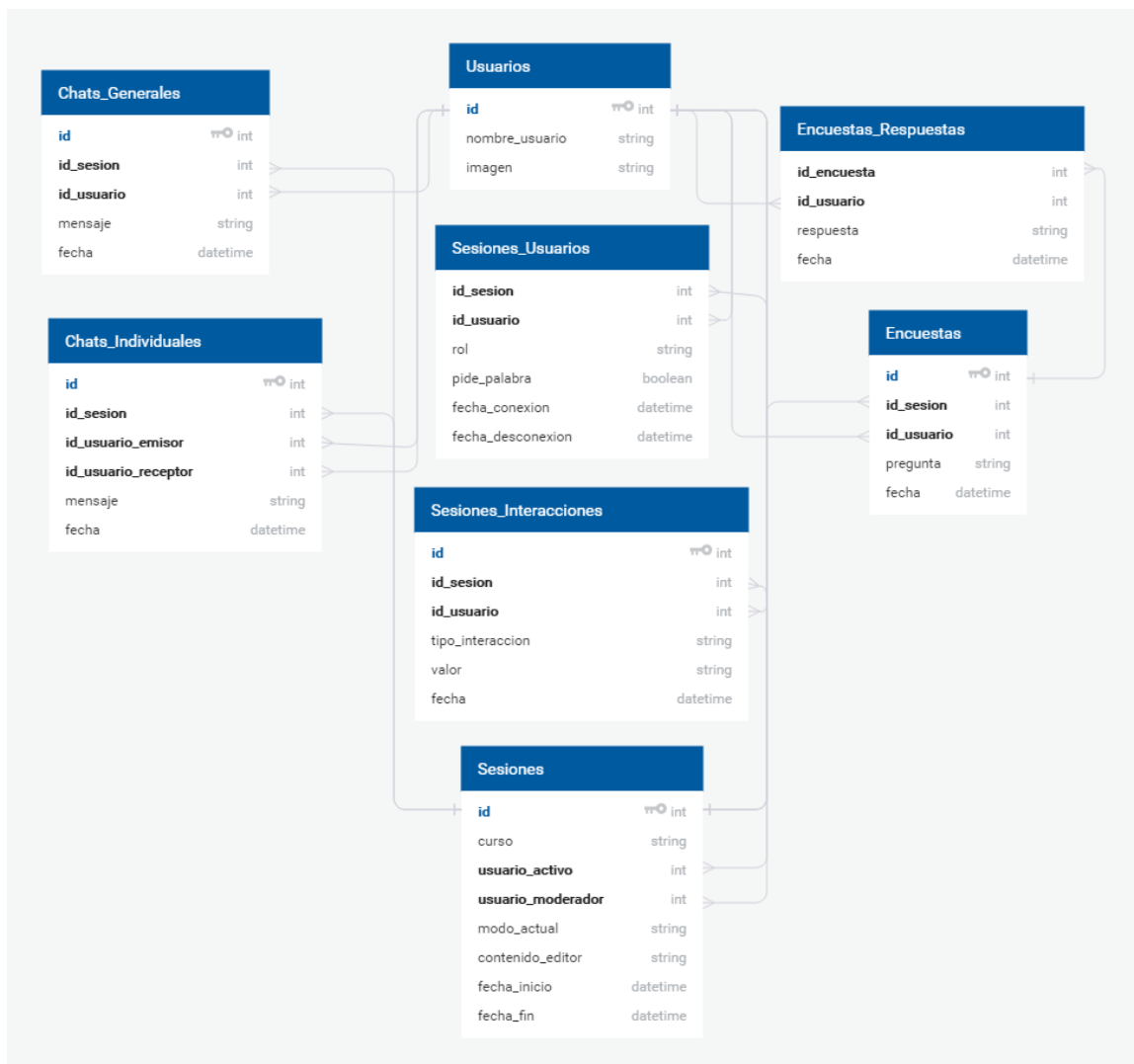


Figura 5.16: Modelo de la base de datos de InfoMeeting.

5.7. API para obtener datos

Para exportar los datos, de manera tal que otros sistemas (particularmente el EVEA IDEAS) puedan utilizarlos y almacenarlos de la manera deseada, InfoMeeting dispone de un *endpoint*, donde se retorna un mensaje json según la estructura que se detalla a en la figura 5.17:

```

sesion: {
  id: 34,
  fechaInicio: 2017-11-15 15:00:00,
  fechaFin: 2017-11-15 15:32:00,
  usuarios: [{
    user1: {
      fecha_conexion: 2017-11-15 15:02:00,
      fecha_desconexion: 2017-11-15 15:22:00,
    },
    user2: {
      fecha_conexion: 2017-11-15 15:04:00,
      fecha_desconexion: 2017-11-15 15:29:00,
    },
  ]},
  encuestas: [
    {
      'pregunta': 'Es 7 un número primo?',
      'fecha': 2017-11-15 15:06:00,
      'id_usuario': 2,
      'respuestas': [
        {
          id_usuario: 4,
          respuesta: true
        },
        {
          id_usuario: 9,
          respuesta: true
        }
      ]
    },
    {
      'pregunta': 'Pertenece Nigeria a Europa?'
      'fecha': 2017-11-15 15:08:00
      'id_usuario': 2,
      'respuestas': [
        {
          id_usuario: 4,
          respuesta: false
        },
        {
          id_usuario: 9,
          respuesta: true
        }
      ]
    }
  ]
},
chats_individuales: [
  {
    id_usuario_emisor: 4 ,
    id_usuario_receptor: 9,
    fecha: 2017-11-15 15:04:00 ,
    Mensaje: "Mensaje de prueba" ,
  },
  {
    id_usuario_emisor: 9,
    id_usuario_receptor: 4 ,
    fecha: 2017-11-15 15:05:00,
    Mensaje: "Los mensajes llegan bien" ,
  },
],
chats_generales: [
  {
    id_usuario: 4,
    fecha: 2017-11-15 15:05:08,
    mensaje: "Hola a todos",
  },
  {
    id_usuario: 9,
    fecha: 2017-11-15 15:05:10,
    mensaje: "Buenos dias, quien es el moderador?",
  },
  {
    id_usuario: 8,
    fecha: 2017-11-15 15:05:15,
    Mensaje: "Hola a todos, yo soy el moderador" ,
  },
  {
    id_usuario: 7,
    fecha: 2017-11-15 15:05:20,
    Mensaje: "Cual es la agenda para hoy?" ,
  },
],
interacciones: [
  {
    id_usuario: 4
    tipo_interaccion: 'pedir control'
    valor: "", // este tipo de interacción no tiene valores
    fecha: 2017-11-15 15:06:20,
  },
  {
    id_usuario: 5
    tipo_interaccion: 'otorgar control'
    valor: 4, // en este caso el valor representa el usuario al que se le dio el control
    fecha: 2017-11-15 15:05:32 ,
  },
  {
    id_usuario: 6
    tipo_interaccion: 'bloquear participante'
    valor: 4, // en este caso el valor representa el usuario al que se bloqueó
    fecha: 2017-11-15 15:05:55,
  },
]
}

```

Figura 5.17: Información expuesta por InfoMeeting.

5.8. Conclusión

En este capítulo se hizo un recorrido por las tecnologías utilizadas para el desarrollo de InfoMeeting. Se comenzó haciendo hincapié en los problemas tecnológicos que presentan las herramientas vistas en el Capítulo 3. Luego se describió la arquitectura de InfoMeeting y se hizo un análisis de las tecnologías más relevantes utilizadas en la implementación de la capa de lógica como Node.js, Express.js, WebSockets y WebRTC. También se mencionaron las tecnologías utilizadas en la capa de presentación. Se describieron las librerías de JavaScript que ayudan al empaquetado de la aplicación y también otras como Bootstrap, Sass y jQuery que facilitaron notablemente el desarrollo.

Además se describió cómo se realiza la persistencia de datos en la aplicación, teniendo como base la entidad sesión, la cual se relaciona con la mayoría de las otras entidades.

Finalmente se introdujo el mecanismo utilizado para la integración con el EVEA IDEAS, y la estructura de datos brindada por InfoMeeting para exponer los datos generados durante las sesiones. En el siguiente capítulo se presentan las pruebas de usabilidad realizadas con InfoMeeting y sus principales resultados.

Capítulo 6

Sesiones de prueba de InfoMeeting

6.1. Introducción

En este capítulo se describen las sesiones de prueba realizadas para la evaluación de InfoMeeting. El proceso de evaluación se realizó en etapas, a partir de diferentes sesiones con distintas intenciones. Las primeras sesiones se orientaron al testeado de la funcionalidad y al cumplimiento de los requerimientos establecidos, a partir de pruebas informales realizadas durante el desarrollo de InfoMeeting entre el tesista, las directoras y otros docentes participantes. Luego, de un extenso período de este tipo de pruebas, se procedió a realizar una sesión formal, en la cual participaron cuatro usuarios, interactuando con la herramienta en un escenario distribuido y heterogéneo en cuanto a los dispositivos desde los que se accedió a InfoMeeting.

El capítulo comienza por describir las sesiones de prueba informales, detallando los hallazgos encontrados en cada una y las mejoras efectuadas al finalizar cada sesión.

Luego, se detalla la planificación de la sesión de prueba final, donde se describen sus objetivos y la metodología utilizada.

Finalmente, se exponen los resultados obtenidos en base a la encuesta realizada al final de la sesión formal.

6.2. Etapa de Sesiones Informales

Durante el desarrollo de InfoMeeting, y en particular, en la última fase del desarrollo, en la que la interfaz de usuario estaba ya consolidada, se realizaron diversas pruebas entre el tesista, las directoras de la tesis, y en algunos casos con otros docentes y usuarios que fueron aportando diversas consideraciones para la mejora de InfoMeeting.

Estas pruebas, en su mayoría, fueron realizadas a distancia, en un ambiente distribuido, y sin un guión para llevar adelante la sesión. Se las considera informales ya que no se definió un protocolo ni una metodología específica para trabajar durante la sesión, sino que consistieron en sesiones exploratorias de la herramienta y donde los participantes descubrieron errores y aportaron sugerencias en vinculación con la usabilidad y la funcionalidad esperada. Durante las primeras pruebas, cuando aún las conexiones WebRTC no estaba bien establecidas, se recurrió a Google Hangouts para comunicarse y comentar los hallazgos obtenidos. Luego de solucionar el inconveniente con las conexiones, se utilizó InfoMeeting no sólo para probar la herramienta en sí, sino también para posteriormente discutir los resultados de tales pruebas y los próximos pasos a seguir.

A continuación se describen las pruebas realizadas y los hallazgos más destacados en cada una.

Sesión / Modalidad	Hallazgos	Acciones Correctivas
1 / Presencial	Se descubrieron <i>bugs</i> al pasar la palabra a un alumno y se detectó que al incorporarse un tercer usuario, dicho usuario (en lugar de ver al moderador), veía al último usuario conectado, produciéndose una cadena inconsistente de conexiones.	Luego de analizar el código, se detectó el problema y se corrigió la manera en que se muestra el <i>stream</i> principal, respetando al usuario que posee la palabra.

<p>2 /</p> <p>Presencial</p>	<p>Se realizó con éxito la conexión entre los dos primeros usuarios, sin embargo al entrar un tercer usuario, éste no podía ver a los participantes.</p> <p>Dado que el tercer participante ingresó utilizando un dispositivo móvil, a diferencia de los dos primeros que utilizaron notebooks, se concluyó que el problema era causado por un bug al intentar conectar un dispositivo móvil.</p>	<p>Se efectuaron unos cambios mínimos en el código y se realizó con éxito una nueva prueba utilizando dos notebooks y un dispositivo móvil.</p> <p>Cabe destacar que los tres dispositivos se encontraban en la misma red.</p>
<p>3 /</p> <p>A distancia</p>	<p>Se realizó nuevamente la prueba, y se produjo el mismo error.</p> <p>Se concluyó que el problema se debía a que los usuarios se conectaban desde diferentes redes. Dicha conclusión se sostuvo, en relación a la manera en que WebRTC realiza las conexiones entre los dispositivos.</p> <p>Se decidió estudiar en profundidad las técnicas de STUN y TURN (ver Anexo A) para conocer más acerca de los posibles problemas de conexión.</p> <p>A su vez, surgió la necesidad de investigar herramientas que permitan debuggear este tipo de aplicaciones, para poder obtener más pistas respecto de la causa del problema.</p>	<p>La herramienta realizaba las conexiones mediante un servidor STUN provisto por google (stun.l.google.com:19302), el cual se sospechaba que no era suficiente para conectar a los participantes en algunos casos.</p> <p>Se buscó realizar una prueba con un servidor TURN. Dado que estos servidores no son disponibilizados al público de manera abierta como los STUN, así se intentó configurar uno manualmente sin éxito. Finalmente, se encontró un servidor disponible por parte de la empresa Avaya.</p> <p>Se configuró la herramienta para que las conexiones WebRTC se realicen mediante el nuevo servidor y se realizó una prueba desde tres redes diferentes sin mayores inconvenientes.</p>

<p>4 / A distancia</p>	<p>La herramienta ya mostraba estabilidad en las conexiones sobre escenarios heterogéneos. Se trabajó sobre las dinámicas que permitía llevar adelante.</p> <p>Se advirtió que la herramienta permitía llevar adelante sesiones tipo Webinar, sin embargo, no fomentaba otro tipo de dinámicas más espontáneas.</p> <p>Además, cuando se tenía la palabra resultaba confuso no ver al otro usuario, haciéndose necesario el uso del chat general para preguntar si los usuarios escuchan bien.</p> <p>La calidad del audio y del video fue muy buena en todo momento.</p> <p>Se descubrieron algunas falencias en la interfaz móvil, como la falta de notificaciones al recibir mensajes, lo que dificultaba la interacción.</p>	<p>Se rediseñó el panel de usuarios para mostrar un video en lugar de un avatar. (Figura A.1)</p> <p>Se agregaron sonidos de notificaciones al recibir nuevos mensajes en el chat general e individual.</p> <p>Se mejoró la interfaz móvil de manera que el chat y la lista de participantes aparezcan colapsados por defecto, evitando sobrecargar de información al usuario.</p>
----------------------------	--	--

<p>5 / A distancia</p>	<p>La aplicación resultó mucho más interactiva luego de aplicar los últimos cambios. El resultado de esta prueba fue muy motivador y optimista, ya que la dinámica de la sesión y la experiencia de usuario mejoraron notablemente.</p> <p>Se detectó un molesto eco entre el video principal y el video local.</p> <p>Las notificaciones del chat individual no se mostraban.</p> <p>El logo de aplicación se veía estirado en dispositivos móviles.</p>	<p>Se solucionó el problema del eco</p> <p>Se implementaron las notificaciones del chat individual.</p> <p>Se mejoró la interfaz sobre móviles, y se ajustó el tamaño del logo.</p>
----------------------------	---	---

<p>6 / A distancia</p>	<p>Durante las pruebas realizadas se notó una mejora notable en la calidad de la comunicación al eliminar el eco que se producía anteriormente.</p> <p>Se encontraron errores al pasar de modo editor a modo video (o viceversa), cuando la acción era ejecutada por un participante que había obtenido la palabra recientemente.</p> <p>Por el lado de la usabilidad y experiencia de usuario, se notó que los colores de las notificaciones no estaban unificados, y se observó la falta de un icono para pedir la palabra al tutor. También se detectó la ausencia de notificaciones cuando hay nuevos resultados en encuestas.</p>	<p>Se eliminó el bug que impedía cambiar de modo cuando un participante había obtenido la palabra.</p> <p>Se agregó un ícono para pedir la palabra al tutor.</p> <p>Se unificaron los colores de las notificaciones y se agregó una nueva notificación para indicar que había una nueva respuesta a una consulta.</p>
----------------------------	--	---

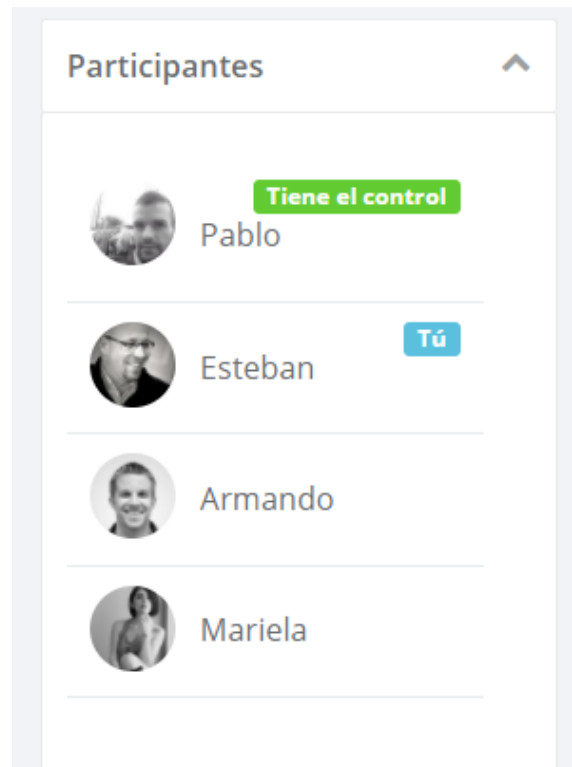


Figura 6.1: Así se veía el panel de participantes antes de la 5ta sesión de prueba. Finalmente, se decidió reemplazar los avatares estáticos por un componente de video.

Luego de estas sesiones, y un largo trabajo de diálogo entre los participantes de esta etapa, se logró contar en InfoMeeting con una funcionalidad acorde a los requerimientos establecidos y con buenas características de usabilidad percibidas.

6.3. Sesión de prueba formal

6.3.1. Introducción

La sesión de prueba formal se realizó un lunes por la tarde, donde cuatro participantes se distribuyeron en distintos espacios físicos, diferentes conexiones de red, y dispositivos, con el fin de probar la herramienta y opinar sobre su usabilidad.

Los participantes fueron convocados para probar la herramienta, pero se les dio una propuesta de trabajo concreta para que la herramienta se utilizará con un fin y no solamente de modo exploratorio. El objetivo que se les propuso fue llevar a cabo una sesión donde los partici-

pantes debían analizar unos fragmentos de código y descubrir errores en el mismo, utilizando la herramienta InfoMeeting. De esta manera, se involucró también el uso del editor de código que ofrece InfoMeeting, y se trabajó en una dinámica que podría ser utilizada a futuro por docentes de la Facultad de Informática.

6.3.2. Reglas de participación y dinámica de la sesión

Para llevar adelante la sesión, se escribió un guión para los participantes, con una serie de reglas y tareas a cumplir durante la sesión. Uno de los participantes tuvo el rol de tutor/coordinador de la sesión (guión de coordinador), mientras que el resto de los participantes tuvo el rol de alumno (guión de participante). Cada participante debía aportar un solo error encontrado en el código publicado por el coordinador, para dar lugar a que todos participen. Si los otros no encontraban errores, entonces sí podían colaborar y dar pistas para que los descubran

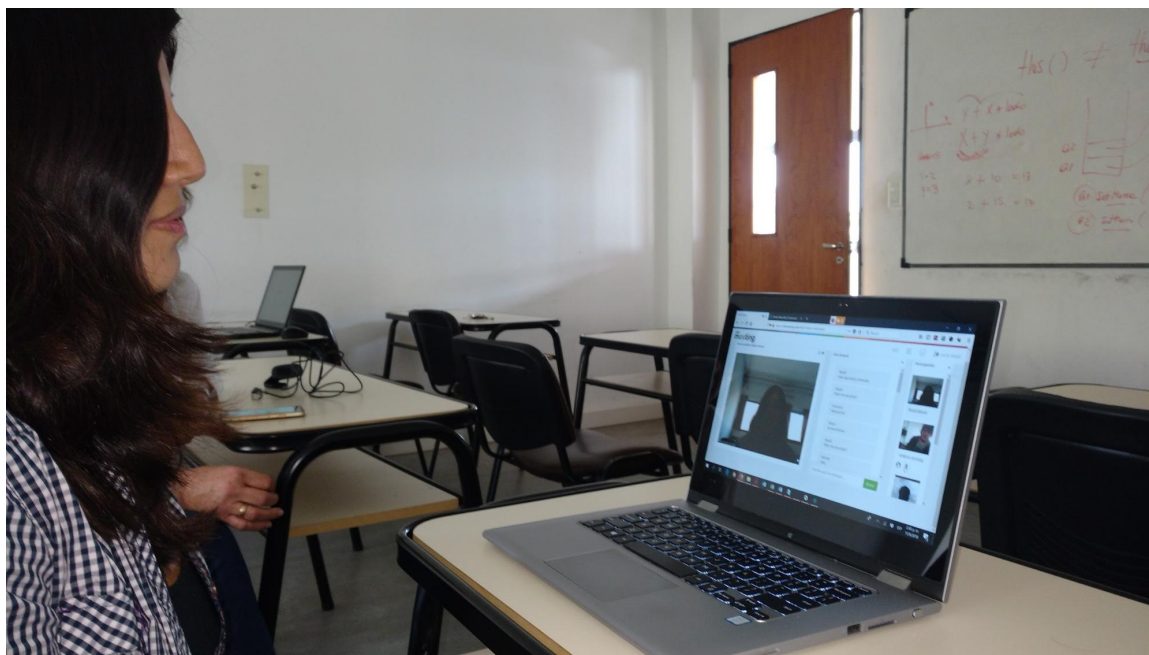


Figura 6.2: Uno de los participantes durante la sesión de prueba.

Dos de los participantes se encontraron distribuidos en las aulas de la Facultad, como se muestra en la Figura A.2 mientras que los otros dos participantes participaron desde sus casas.

Los guiones para cada tipo de participante pueden verse en el Anexo B de este trabajo.

La sesión tuvo una duración de 54 minutos.

6.3.3. Problemas encontrados durante la sesión

La sesión se llevó a cabo sin mayores inconvenientes, aunque uno de los participantes tuvo problemas para entrar a la herramienta y mantener una conexión estable con el resto. Dicho participante, volvió a conectarse unos minutos más tarde desde otra red, y pudo mantener la conexión durante toda la sesión.

Se notó una cierta lentitud en la propagación de caracteres al escribir en el editor de código, aunque esto no afectó a la resolución de los ejercicios.

6.3.4. Encuesta

Al finalizar la sesión de prueba, los participantes realizaron una encuesta utilizando la herramienta Google Forms, la cual permite completar formularios online y generar estadísticas a partir de las respuestas de los participantes.

Dicha encuesta estuvo dividida en cinco partes: datos personales; aspectos generales de InfoMeeting; aspectos de la interfaz de InfoMeeting; performance de la herramienta; y una sección con preguntas específicas para el coordinador de la sesión.

A continuación se analizarán los resultados obtenidos. El lector puede encontrar la lista completa de preguntas en el Anexo C.

Parte 1 - Datos Personales

La primera parte de la encuesta permitió obtener un perfil de los participantes. Todos ellos pertenecen al área de desarrollo de sistemas, con edades entre 24 y 31 años. A su vez, estaban familiarizados con la mayoría de las herramientas de comunicación general. En la figura 6.3, se puede visualizar que los 4 participantes utilizan WhatsApp y Skype, y en menor medida, Slack y Hangout.

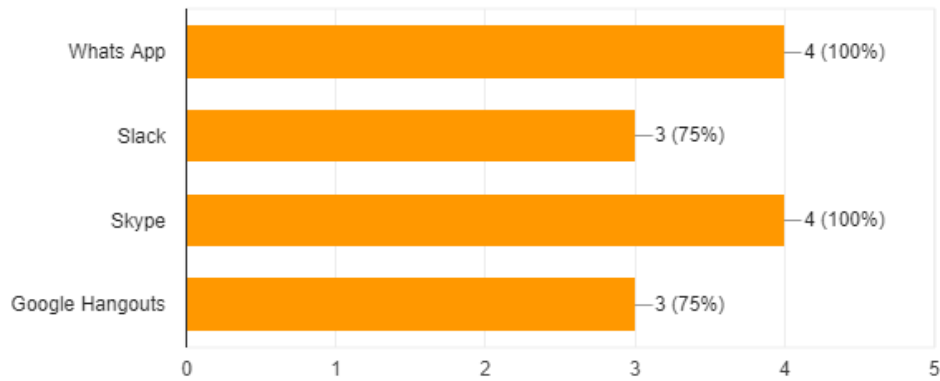


Figura 6.3: Respuestas a la pregunta *Indique cuales de las siguientes herramientas de comunicación de propósito general ha utilizado.*

Por otro lado, como se observa en la figura 6.4 al preguntar por su conocimiento acerca de las herramientas analizadas en el capítulo 3, se descubrió que WebEx era la más conocida por los participantes.

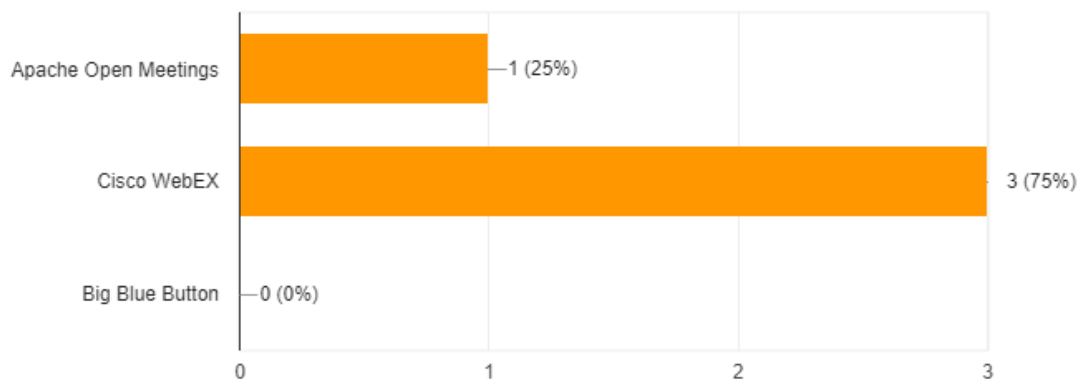


Figura 6.4: Respuestas a la pregunta *Indique si conoces alguna de estas herramientas que fueron utilizadas como inspiración para crear InfoMeeting.*

Parte 2 - Aspectos Generales

La segunda parte de la encuesta tuvo como objetivo relevar algunos aspectos generales de la aplicación. La tabla 6.2 muestra las respuestas a la pregunta *¿Qué fue lo que más les gustó de InfoMeeting?*.

Tales respuestas, muestran que 3 de los 4 participantes se vieron interesados en el editor de código. Las figuras 6.5 y 6.6, muestran el interés de los participantes por estas funcionalidades.

Participante	Respuesta
1	El editor interactivo
2	La interfaz minimalista, y el poder incorporarse a la sala dinámicamente
3	Poder ver y editar código en tiempo real, mientras se mantiene una comunicación audiovisual con el resto de los participantes
4	El hecho de poder entablar conversaciones y crear o modificar código en una misma plataforma

Tabla 6.2: Respuestas a la pregunta *¿Qué fue lo que más les gustó de InfoMeeting?*.

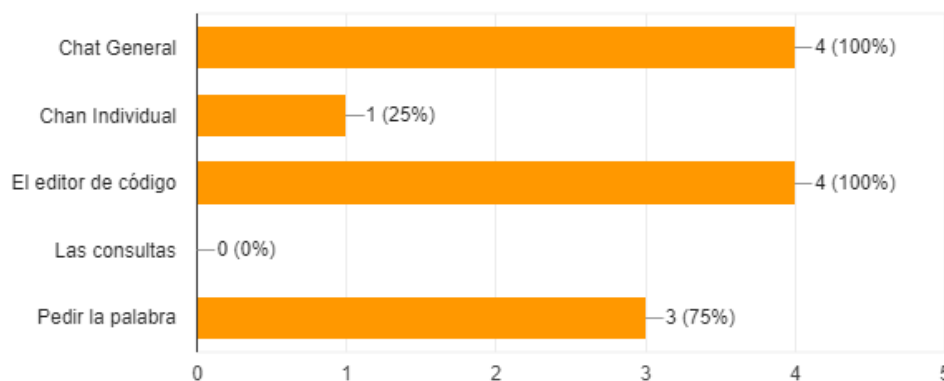


Figura 6.5: Respuestas a la pregunta *¿Qué funciones le parecieron mas útiles?*

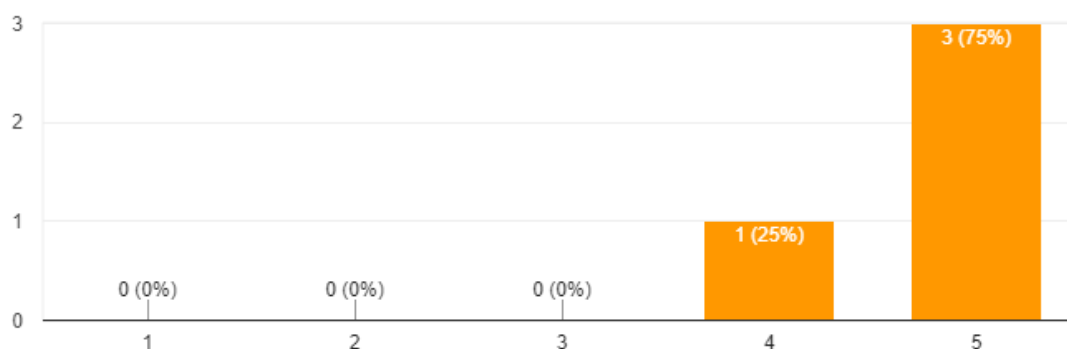


Figura 6.6: Respuestas a la pregunta *¿Qué tan útil le pareció el editor de código?*

Con respecto a los aspectos que menos le gustaron a los participantes, las respuestas obtenidas pueden verse en la tabla 6.3.

Analizando las respuestas se puede inferir que uno de los participantes se unió a la sesión desde una red lenta. Por otro lado, las respuestas 2 y 3 son muy valiosas para incorporar como mejoras en el desarrollo de la herramienta.

Participante	Respuesta
1	Las interferencias de video y acoples de audio
2	Algunos <i>bugs</i> menores, y no tener la opción de deshabilitar el video y/o audio.
3	La funcionalidad de pedir la palabra podría mejorarse. Resaltar el usuario que pide la palabra o mostrar un mensaje <i>fixed</i> cuando se participa desde celular.
4	Nada en particular

Tabla 6.3: Respuestas a la pregunta *¿Qué fue lo que menos le gustó InfoMeeting?*.

Participante	Respuesta
1	- Posibilidad de interacción sin vídeo, para ver el código y poder hablar. - Algún color que diferencie a quien tiene la palabra en la lista de integrantes de la derecha, para mejorar la visual cuando está el código. - Que se pueda pedir la palabra y devolver la palabra desde un icono más llamativo o quizá ubicado al costado de la pantalla donde esta el editor para que sea más accesible.
2	La posibilidad de deshabilitar el video y/o audio para alivianar el consumo de red.
3	Creo que la herramienta tiene las funcionalidades necesarias.
4	Se podría sumar funcionalidades como compartir archivos, o una versión paralela a la actual en la que se puede editar código entre varias personas al mismo tiempo, y que la persona que crea la meeting decida en cual version trabajar. Por decir algo. Así como esta esta muy buena la pagina.

Tabla 6.4: Respuestas a la pregunta *¿Qué funcionalidad le agregaría a la herramienta?*.

Otra de las preguntas que aportó información valiosa para mejoras y futuras características fue *¿Qué funcionalidad le agregaría a la herramienta?* cuyas respuestas pueden verse en la tabla 6.4.

Resulta particularmente interesante el ítem 2, donde en casos de conexiones lentas, la opción de deshabilitar el video podría ser determinante para el desarrollo de la sesión.

Finalmente, los participantes coincidieron en que la herramienta puede ser un complemento apropiado para la comunicación entre alumnos y docentes. Esto puede observarse en la figura 6.7.

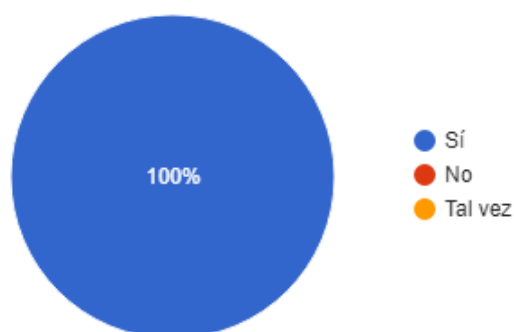


Figura 6.7: Respuestas a la pregunta *¿Considera que InfoMeeting puede ser un complemento apropiado para la comunicación entre alumnos y docentes durante el dictado de un curso?*.

Parte 3 - Acerca de la interfaz y la experiencia de usuario

La tercera parte tuvo como objetivo relevar la satisfacción de los usuarios respecto de la interfaz de usuario y detectar posibles falencias en la misma. Como se observa en la figura 6.8, al preguntar qué tan intuitiva les resultó la interfaz, el 75 % (3 participantes) consideró que la interfaz es muy intuitiva, mientras que el restante (25 %) la considero intuitiva.

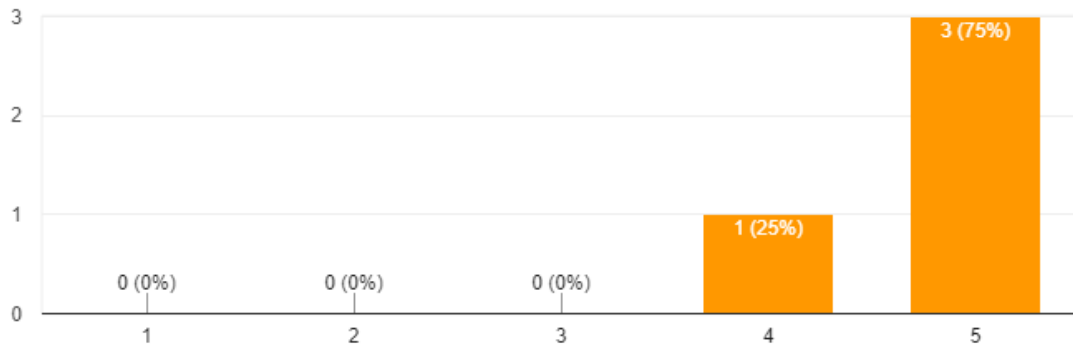


Figura 6.8: Respuestas a la pregunta *¿Qué tan intuitiva le resultó la interfaz (estructura, organización, etc)?*. El rango de respuestas iba desde poco intuitiva hasta muy intuitiva.

Al preguntar si acerca de las fuentes utilizadas resultaron adecuadas (figura 6.9), el 75 % de los participantes consideró que eran muy adecuadas, mientras que el restante (25 %) consideró que eran adecuadas.

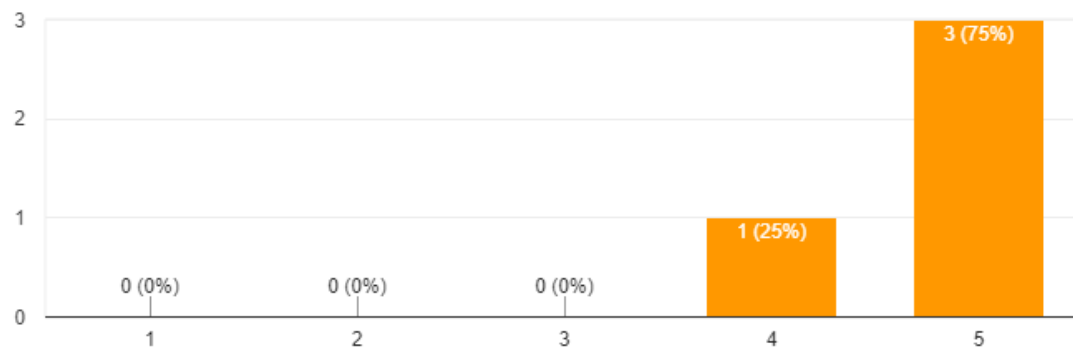


Figura 6.9: Respuestas a la pregunta *¿Considera que las fuentes utilizadas, así como el tamaño de las mismas es adecuada?*. El rango de respuestas iba desde poco adecuadas hasta muy adecuadas.

Respecto de los colores elegidos, se preguntó a los usuarios si consideraban que el contraste

era apropiado y favorecía la legibilidad. La figura 6.10 muestra que el 100 % de los participantes estaban de acuerdo.

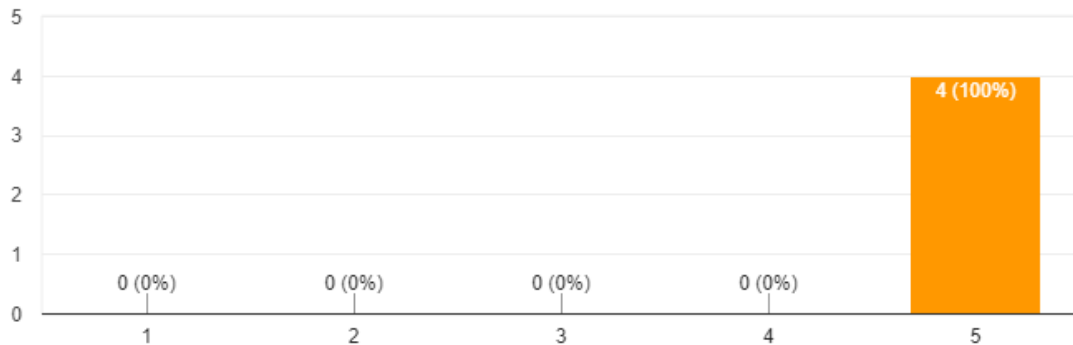


Figura 6.10: Respuestas a la pregunta *¿Considera que los colores utilizados tienen un buen contraste y favorecen la legibilidad?*. El rango de respuestas iba desde muy en desacuerdo hasta muy de acuerdo.

Respecto de los iconos elegidos (los cual fueron obtenidos de versiones gratuitas de la librería de iconos Font Awesome y Pixeden), se observa en la figura 6.11 que cumplieron las expectativas.

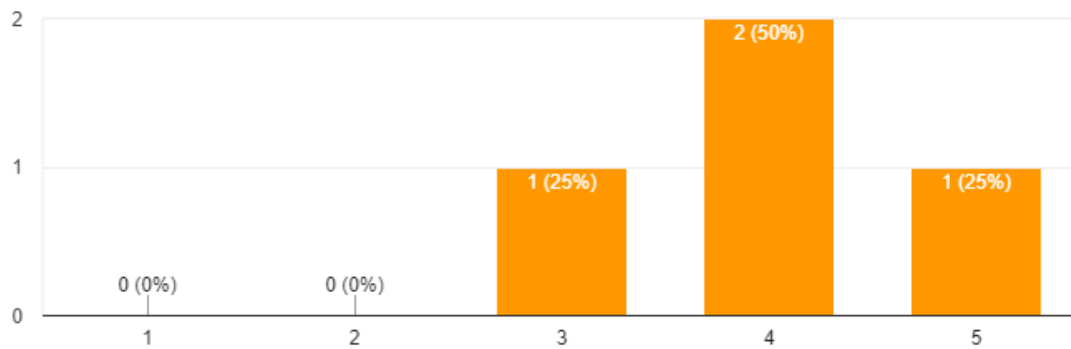


Figura 6.11: Respuestas a la pregunta *¿Qué tan intuitivos le resultaron los íconos?*. El rango de respuestas iba desde poco intuitivos hasta muy intuitivos.

Respecto de las notificaciones sonoras al recibir mensajes, parece haber un consenso en cuanto a que resultan apropiadas como se observa en la figura 6.12. No obstante, uno de los usuarios, como se verá en la tabla 6.5 pregunta, experimentó problemas de acople con tales notificaciones.

Participante	Respuesta
1	Cuando otro usuario recibe notificaciones, el sonido se capta a través del micrófono y genera acople.
2	Un señalamiento sobre quién tiene la palabra en el momento.
3	Resaltaría al usuario que tiene la palabra con algún color llamativo.
4	Enfatizar más quien tiene la palabra

Tabla 6.5: Respuestas a la pregunta *¿Qué aspectos de la interfaz cree que pueden mejorarse?*.

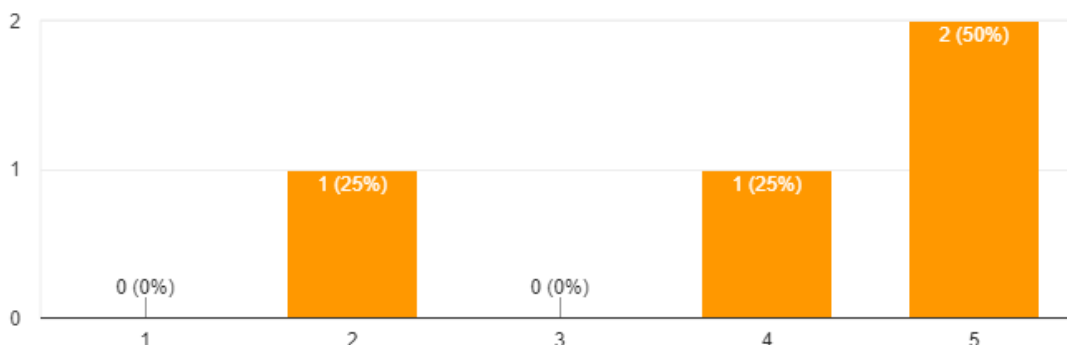


Figura 6.12: Respuestas a la pregunta *¿Considera que las notificaciones sonoras al recibir mensajes son apropiadas?*. El rango de respuestas iba desde *muy en desacuerdo* hasta *muy de acuerdo*.

Por último se le pidió a cada participante que escriban libremente *¿Qué aspectos de la interfaz cree que pueden mejorarse?*. La tabla 6.5 muestra los resultados, donde se puede observar que, la mayoría de los participantes, concuerdan en que no resulta claro quién tiene la palabra cuando la herramienta está en modo editor de código.

6.3.5. Parte 4 - Acerca de la performance

La última parte de la encuesta tuvo como objetivo realizar algunas mediciones respecto de la performance de la aplicación, así como también indagar acerca de los medios utilizados para acceder a la aplicación.

Como se observa en la pregunta *¿Cómo le resultó el tiempo de carga y de respuesta la aplicación?* (Figura 6.13), el grado de satisfacción fue bueno. Esto resulta reconfortante, ya que se pensaba que, debido a la naturaleza de la aplicación, el uso constante de la red impactaría en el tiempo de respuesta.

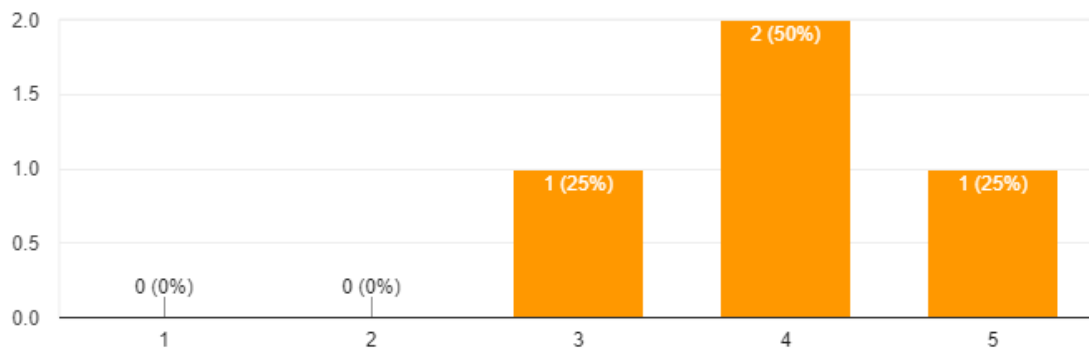


Figura 6.13: Respuestas a la pregunta *¿Cómo le resultó el tiempo de carga y de respuesta la aplicación?*. El rango de respuestas iba desde *muy lento* hasta *muy rápido*.

Se obtuvieron respuestas similares al preguntar por la calidad del audio y del video, como se puede ver en la figuras 6.14 y 6.15. Esto resulta interesante, ya que las respuestas fueron buenas, a pesar de que algunos de los participantes utilizaron conexiones a internet estándares.

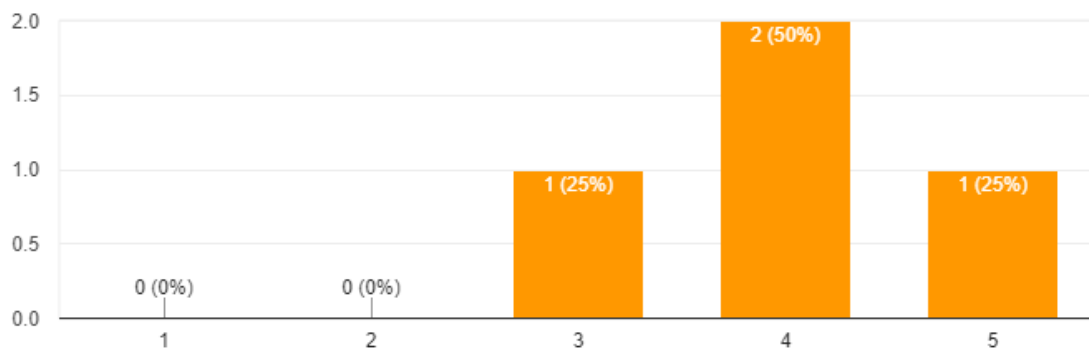


Figura 6.14: Respuestas a la pregunta *¿Cómo notó la calidad del video?*. El rango de respuestas iba desde *muy baja* hasta *muy alta*.

Participante	Respuesta
1	Windows 10 y Firefox
2	Windows y Chrome
3	Android 7, Chrome
4	Android, Google Chrome

Tabla 6.6: Respuestas a la pregunta *¿Qué Sistema Operativo y navegador utilizó?*.

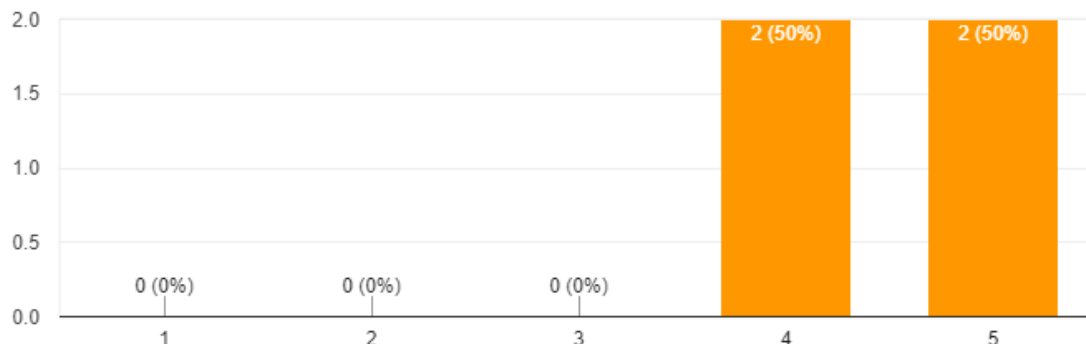


Figura 6.15: Respuestas a la pregunta *¿Cómo notó la calidad del audio?*. El rango de respuestas iba desde *muy baja* hasta *muy alta*.

Luego se les preguntó a los participantes acerca de los dispositivos (Figura 6.16) y del software (Tabla 6.6) utilizados para la prueba.

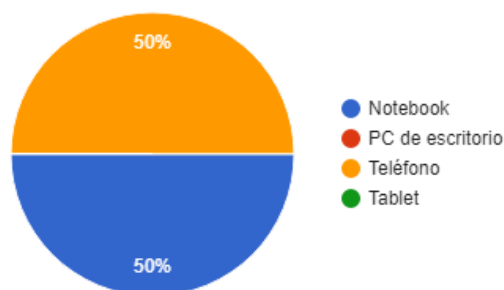


Figura 6.16: Respuestas a la pregunta *¿Qué dispositivo utilizó?*.

Como se observa en las respuestas, hubo una distribución equitativa en el uso de notebooks y smartphones. Resulta de vital importancia para InfoMeeting, el hecho implementar en su versión móvil las mismas características que en su versión escritorio, dado el acelerado y constante crecimiento en el uso de dispositivos móviles que se viene dando hace varios años¹.

¹<https://wearesocial.com/special-reports/digital-in-2017-global-overview>

Finalmente se les preguntó a los participantes acerca de la conexión a internet utilizada.

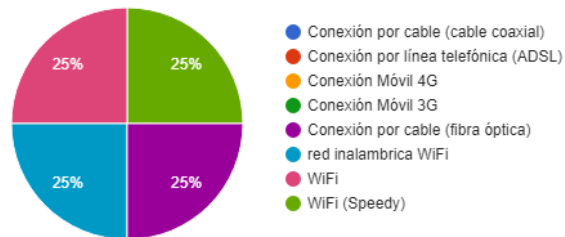


Figura 6.17: Respuestas a la pregunta *¿Qué tipo de conexión a internet utilizó?* .

Si bien esta pregunta tuvo una lista de opciones disponibles, los participantes también podían escribir libremente eligiendo la opción “otra”. Un sondeo posterior a la encuesta determinó que en realidad el 50 % de los participantes utilizó una conexión de fibra óptica mientras que el otro 50 % una conexión estándar de ADSL.

6.3.6. Anexo para el Tutor

El participante designado para llevar a cabo la sesión tuvo un apartado exclusivo con cuatro preguntas, cuya intención fue la de medir la experiencia de usuario al controlar la sesión. Las preguntas y las respuestas pueden observarse en la figura 6.18

¿Cuán fácil le resultó llevar a cabo la sesión?

1 2 3 4 5

Muy difícil Muy fácil

¿Qué tan útiles encontró las notificaciones?

1 2 3 4 5

Poco útiles o escasas Muy útiles

¿Qué tan fácil le resultó darle la palabra a otro usuario?

1 2 3 4 5

Muy difícil Muy fácil

Figura 6.18: Preguntas y respuestas para el tutor. Las escalas utilizadas iban desde *Muy difícil* a *Muy fácil* en la primer y tercera pregunta, mientras que para la segunda se uso un rango desde *Poco útiles o escasas* hasta *Muy útiles*.

Las respuestas obtenidas son muy alentadoras, ya que la moderación es un aspecto clave de InfoMeeting. Tal aspecto demandó mucho tiempo del proceso, tanto para pensar en los casos de uso como en la interfaz de usuario apropiada para implementar dichos casos.

Por último se le preguntó: ¿Qué funciones le agregaría a la herramienta para facilitar o potenciar la moderación de las sesiones? La respuestas obtenida fue:

Una visualización para listar de forma más compacta a los participantes.

La observación resulta valiosa, puesto que ayudaría a minimizar la interfaz reduciendo espacio que no se necesita.

6.4. Conclusión

En este capítulo se describieron las sesiones, formales e informales, que se llevaron a cabo durante el desarrollo de la herramienta. Se comenzó describiendo las pruebas informales, donde se detectaron varias cuestiones propias del trabajo en entornos distribuidos y heterogéneos, que habían pasado desapercibidos durante el desarrollo, dado que las pruebas hechas hasta el momento se realizaron en un mismo dispositivo. El hecho de incorporar nuevos dispositivos y

nuevas redes, fue clave en la detección de estos *bugs*.

Una vez solucionados los *bugs*, las sesiones fueron muy enriquecedoras para la obtención de *feedback* relacionado con la experiencia de usuario. Esto llevó a rediseñar el panel de usuarios y a incorporar notificaciones visuales y sonoras.

Luego se describió la sesión formal, la cual consistió en una prueba distribuida con cuatro usuarios que desconocían la herramienta, con la consigna de utilizar el editor de código de InfoMeeting para detectar errores en algoritmos, y probar las funcionalidades generales de la aplicación. Los participantes realizaron una evaluación de la herramienta al finalizar su participación.

Finalmente se expusieron los resultados de la evaluación, los cuales fueron muy positivos y alentadores, ya que validaron el propósito y la intención por el cual fue creada la herramienta. Por otro lado, se obtuvo un valioso *feedback* para mejorar algunas características visuales, como por ejemplo, destacar al usuario que tiene la palabra en el panel de usuarios. Dicho *feedback* también permitió incorporar nuevas funcionalidades, como es el caso de activar y desactivar el video, para reducir el ancho de banda en conexiones de baja velocidad, lo que ya ha sido implementado.

Capítulo 7

Conclusión y trabajos futuros

7.1. Introducción

En este capítulo se resumen los aportes del trabajo realizado a lo largo de esta tesina. Se presentan algunas conclusiones respecto del proceso de desarrollo y se mencionan los posibles caminos para continuar con el desarrollo de la herramienta y su validación.

7.2. Aportes realizados y Conclusiones

Se hizo una revisión de algunos trabajos sobre el uso de la comunicación asincrónica y sincrónica en escenarios educativos y se profundizó en esta última mencionando algunos casos particulares, como los de 3C en China, Wimba en Estados Unidos y PEDCO/Skype en Argentina.

También se analizaron algunas herramientas actuales de comunicación sincrónica, dos de ellas open source (Apache Open Meetings y Big Blue Button) y otra que se ofrece en la modalidad de software como servicio (WebEx). El análisis técnico de las dos primeras, detectó que las tecnologías subyacentes resultaron desactualizadas y pueden suponer una barrera de entrada, especialmente para los desarrolladores más jóvenes, desalentando la participación en su evolución y mantenimiento. Al mismo tiempo, se trata de herramientas generales y no específicamente diseñadas para escenarios educativos.

Tal hallazgo motivó una búsqueda de tecnologías más modernas para la implementación de comunicación en tiempo real, lo cual llevó a investigar opciones tales como WebSockets y

WebRTC. Al mismo tiempo, surgió la idea de crear una herramienta que pueda ser utilizada para la comunicación sincrónica entre alumnos y docentes, inspirándose en las aplicaciones mencionadas, y utilizando tecnologías más modernas.

En efecto, se procedió con el diseño e implementación de la herramienta, la cual fue llamada InfoMeeting, cuyo nombre es un juego de palabras entre Info, por estar pensada para utilizar por alumnos y docentes de la Facultad de Informática; y Meeting, término que significa encuentro en inglés.

Las etapas de diseño, implementación y validación de InfoMeeting, resultaron en un proceso de aprendizaje rico, ya que se encontraron barreras que debieron ser superadas a partir del estudio de diferentes tecnologías y estrategias. Asimismo, se llevó a cabo una sesión de prueba, con cuatro usuarios que desconocían la herramienta. Esta sesión arrojó resultados satisfactorios que alientan a continuar con el desarrollo de InfoMeeting.

La herramienta fue bien aceptada, tanto del lado del tutor como de los alumnos participantes y fue considerada como un complemento útil para integrar en un EVEA. La dinámica de interacción propuesta durante la sesión, con el concepto de pedir la palabra resultó natural y fue llevada a cabo sin mayores inconvenientes. Tal aceptación resultó por demás gratificante.

El editor de texto, el cual fue pensado originalmente como un complemento extra, fue muy bien recibido y se lo consideró como una funcionalidad muy útil y original. Asimismo, resulta especialmente apropiado para utilizar en cursos donde se trabaja con algoritmos.

Las tecnologías seleccionadas para implementar InfoMeeting son muy interesantes y hay buena documentación para trabajar con ellas. Esto permitirá que se pueda extender este trabajo incorporando nuevas funcionalidades en el futuro.

Personalmente, aprendí mucho acerca del uso de WebSockets y WebRTC, así como de programación distribuida en general. Por otro lado, fue muy gratificante y enriquecedor el trabajo conjunto con los directores y con el equipo de IDEAS.

No obstante, he tenido momentos de frustración, dado que el desarrollo de una aplicación distribuida y para ser utilizada en escenarios heterogéneos agregó una complejidad al momento de su testeo y posterior validación.

7.3. Líneas de trabajo futuro

En referencia a los comentarios recibidos durante las pruebas, las herramientas analizadas en el capítulo 2 y aquellas características que debido al limitado tiempo del trabajo no han podido ser implementadas, se considera que estas son posibles líneas de trabajo futuro:

Por el lado del editor de código se propone:

- Extender el funcionamiento del editor para propagar eventos de scroll, foco y selección de texto.
- Dar soporte para el resaltado de sintaxis de múltiples lenguajes de programación y permitir elegir el lenguaje que se quiera utilizar.
- Permitir no sólo la escritura, sino también la ejecución de código.
- Incorporar un mecanismo de trazabilidad para saber qué partes del código han sido escritas por un determinado usuario.

Por el lado del módulo de consultas:

- Extender su capacidad para que puedan realizarse encuestas de tipo opción múltiple.
- Mejorar la visualización de resultados y brindar la posibilidad de exportarlos.

Otras líneas de trabajo:

- Definir una capa de integración que mejore la comunicación con otros sistemas.
- Crear un panel de control donde se pueda monitorear el estado de la herramienta y guardar un log de errores.
- Implementar el *frontend* de la aplicación utilizando una tecnología más moderna basada en componentes como React o Vue js.

Al mismo tiempo resta aún finalizar el proceso de integración con IDEAS, aunque ya se ha preparado esto y sólo resta testearlo.

Anexos

Anexo A

WebRTC

En este anexo se describe la tecnología WebRTC, que es la base tecnológica sobre la cual se implementa la plataforma InfoMeeting.

Web Real-Time Communication (WebRTC) es una tecnología que permite una comunicación entre navegadores, mediante la cual es posible el envío de audio y video, compartir la pantalla o compartir archivos, sin la necesidad de usar *plugins* internos o externos. De esta manera, WebRTC permite establecer rápidamente conexiones *peer-to-peer* entre navegadores, algo que, en caso de querer construirse desde cero, necesitaría de una gran cantidad de *frameworks* y librerías para lidiar con problemas como la pérdida de datos, la caída de las conexiones o el NAT traversal. Con WebRTC, todas esas funcionalidades ya vienen implementadas dentro del navegador.

Esta tecnología fue desarrollada por Google y lanzada como open source en mayo de 2011¹. Luego la IETF continuó con el trabajo ayudando a la estandarización de los protocolos, mientras que la W3C se ocupa de las APIs del navegador. Actualmente los principales navegadores como Chrome, Firefox y Safari cuentan con soporte nativo para WebRTC.

En comparación, otros métodos de comunicación entre navegadores usan un modelo cliente-servidor, en el cual todo el tráfico pasa por el servidor para que dos clientes se comuniquen. Con WebRTC, en cambio, el servidor solo se usa para establecer las conexiones iniciales entre los *peers*, permitiendo así, que el tráfico circule directamente entre ellos.

¹<http://lists.w3.org/Archives/Public/public-webrtc/2011May/0022.html>

A.1. Arquitectura

La tecnología WebRTC presenta una arquitectura compleja. Dentro de la misma, como se ve en la figura A.1 podemos observar tres APIs:

- Una API para desarrolladores web, la cual provee los objetos `RTCPeerConnection`, `RTCDataChannel`, etc
- Una API para desarrolladores de navegadores
- Una API desde donde los distintos navegadores pueden customizar ciertos componentes reescribiéndolos.

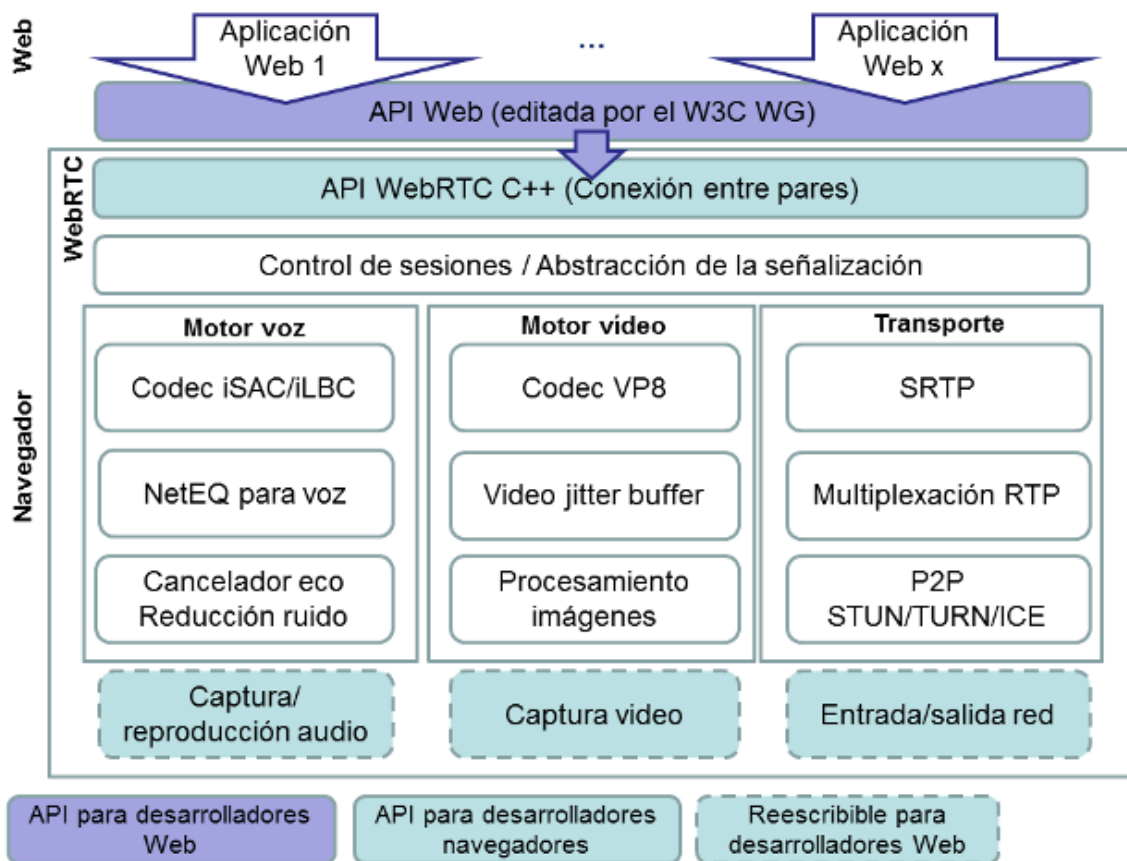


Figura A.1: Arquitectura de la tecnología WebRTC

El gráfico muestra además la presencia de dos motores, el de audio y video. Estos módulos de software permiten la captura y codificación de datos desde dispositivos como la cámara o el micrófono.

También se observan componentes relacionados con el transporte de los datos, cuya función es la de establecer conexiones entre diferentes tipos de redes y dispositivos.

Para el desarrollo de una herramienta como InfoMeeting, el componente más importante de esta tecnología es la API para desarrolladores web. Sin embargo, en lugar de estudiar dicha API, se describirán con mayor detalle algunos de los otros componentes, con el objetivo de entender mejor el contexto sobre el cual se desarrolla el software WebRTC, y para estar al tanto de los problemas de conexión, seguridad y privacidad que pueden darse debido al uso de NAT.

La comunicación *peer-to-peer* en tiempo real es el pilar principal de la tecnología WebRTC. Para lograr la comunicación entre dos *peers* (o clientes) a través del navegador, estos necesitan primero visitar un sitio web donde puedan negociar los datos de su conexión. Luego, los navegadores deben acordar iniciar una comunicación y deben saber cómo localizarse a través de internet, pasando a través de protecciones de seguridad y de *firewalls*. Por último deben saber cómo transmitir los datos en tiempo real. A continuación, describiremos algunos de los componentes tecnológicos utilizados para efectuar dicha localización y transporte de los datos.

A.2. Session description protocol

Session Description Protocol (SDP) es el standard utilizado para establecer las conexiones entre *peers*, permitiendo describir los parametros de inicialización. SPD no es el encargado de entregar los datos, sino la descripción y las propiedades de los mismos. El propósito de SDP es describir sesiones multimedia para anuncios, invitaciones y otras formas de iniciación de sesiones. Una descripción de sesión SDP (Figura A.2) incluye la siguiente información acerca del medio:

- El tipo del medio (video, audio, etc.)
- El protocolo de transporte (RTP/UDP/IP, H.320, etc.)
- El formato del medio (H.261 video, MPEG video, etc.)

SDP incluye además la siguiente información de conexión:

- El tipo de red (usualmente IN para internet)
- El tipo de dirección (IP4, IP6)

- La dirección ip o host que enviará los paquetes

```
v = 0
o = mhandley2890844526 2890842807 IN IP4 126.16.64.4
s = SDP Seminar
i = A Seminar on the session description protocol
u = http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
e = mjh@isi.edu(Mark Handley)
c = IN IP4 224.2.17.12/127
t = 2873397496 2873404696
a = recvonly
m = audio 49170 RTP/AVP 0
m = video 51372 RTP/AVP 31
m = application 32416udp wb
a = orient:portrait
```

Figura A.2: Un ejemplo de mensaje SDP. El parámetro c describe los datos de la conexión.

A.3. Network Address Translation

NAT es un técnica que realiza una correspondencia entre direcciones internas de una red con direcciones externas. De esa manera permite conectar varios dispositivos locales con una conexión externa. Cada vez que un dispositivo local emite datos hacia fuera de la red, NAT traduce la ip del dispositivo a la ip pública de la red (Figura A.3). Cuando el paquete es recibido desde fuera de la red, NAT lo traduce inversamente al dispositivo correcto de la red local.

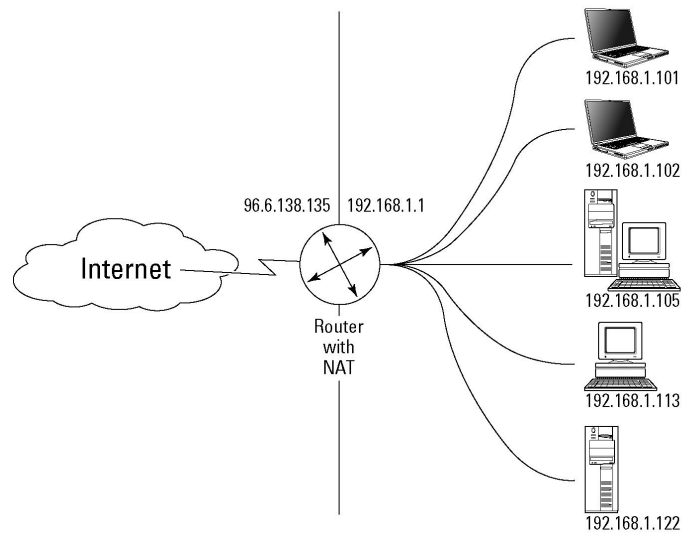


Figura A.3: NAT permite comunicar varios dispositivos locales con direcciones ip externas

A.4. Atravesando barreras

En un enfoque ideal, cada aplicación WebRTC debería ser capaz de aprender su dirección IP externa, la cual podría intercambiar con otros dispositivos para comunicarse directamente. Pero en la realidad, la mayoría de los dispositivos se encuentran detrás de una o más capas de NAT, *firewalls*, *proxys* y programas antivirus, los cuales pueden bloquear ciertas direcciones, puertos y protocolos. A continuación se describen dos alternativas para superar estos inconvenientes.

A.4.1. STUN

La tecnología STUN (*Session Traversal Utilities for NAT*) ayuda a los protocolos a atravesar NAT y le permite a un dispositivo determinar la ip y el puerto que le fue asignado por NAT. También puede utilizarse para chequear conectividad entre dos *endpoints*, o como un protocolo *keep-alive* para mantener enlaces NAT. La figura A.4 muestra una representación de esta técnica.

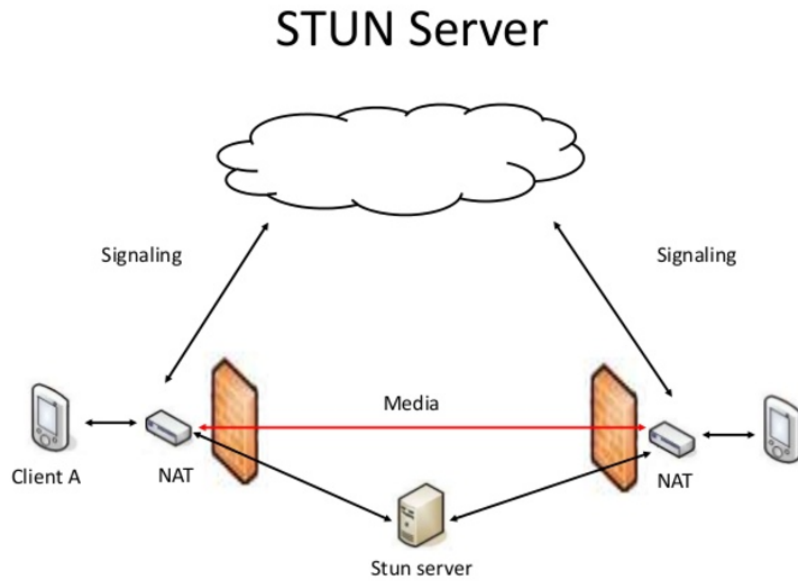


Figura A.4: Comunicación utilizando un Servidor STUN

El objetivo de un STUN server es chequear la ip y el puerto de las peticiones entrantes de aplicaciones corriendo bajo NAT y retornar esa dirección como respuesta. Las aplicaciones WebRTC pueden usar STUN para descubrir la ip y puerto desde una perspectiva pública. Esto le permite a un nodo obtener su propia dirección pública y enviarla a otro para establecer un enlace directo. De acuerdo a las estadísticas de uso, el 92 % de las peticiones realizadas a un servidor STUN fueron exitosas ².

A.4.2. TURN

Si un host está ubicado detrás de un *proxy* o de un *firewall*, STUN fallará al crear una conexión p2p. Cuando esto suceda, existe un último recurso llamado TURN (*Traversal Using Relays around NAT*). Los servidores TURN tienen una dirección pública, para que puedan ser alcanzados por los dispositivos, aun si el dispositivo está detrás de un *firewall* o un *proxy*. Pero a diferencia de los servidores STUN, estos tendrán una carga de tráfico grande, lo cual hará que estén más ocupados que los STUN. La figura A.5 muestra una representación de esta técnica.

²https://developers.google.com/talk/libjingle/important_concepts

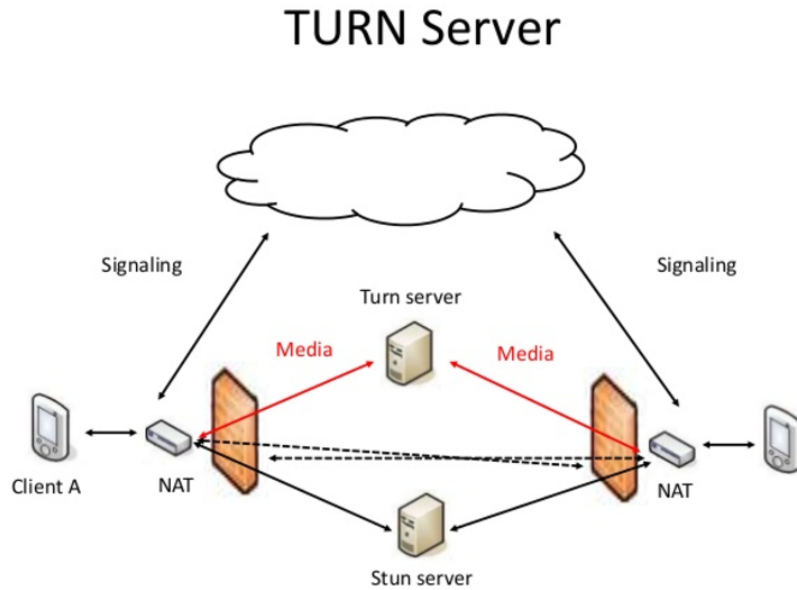


Figura A.5: Comunicación utilizando un Servidor TURN

En este caso la comunicación no será *peer-to-peer*, pero se permitirá la conexión de una mayor cantidad de dispositivos.

A.5. ICE

ICE o *Interactive Connectivity Establishment* es un *framework* que permite lidiar con los problemas de NAT para poder establecer una comunicación interactiva entre dos dispositivos. Una instalación ICE cuenta con dos *endpoints* (los *peers*) que se quieren comunicar. Los *peers* pueden comunicarse indirectamente con un *signaling server* donde pueden intercambiar mensajes SDP. En el comienzo del *setup*, los *peers* no conocen su propia topología con lo cual no saben si están detrás de uno (o múltiples) NATs. ICE ayuda a los *peers* a descubrir información acerca de su topología para que puedan encontrar uno o más caminos por los cual puedan comunicarse. ICE intentará probar todos los caminos posibles en paralelo y elegir el más eficiente. Primero intenta conectar usando la ip del host obtenida desde el sistema operativo y la tarjeta de red, la cual fallará si estuviera detrás de NAT. Luego obtendrá la ip externa usando STUN y si esta también fallara, se reintentará con un servidor TURN.

A.6. SCTP

Todas las aplicaciones WebRTC utilizan el protocolo UDP (*User Datagram Protocol*) como protocolo base de transporte. La figura A.6 muestra la pila de protocolos utilizados en una comunicación WebRTC. UDP permite el envío de paquetes entre computadores en una red ip mediante un mecanismo muy básico, lo cual lo convierte en un medio muy eficiente pero que no garantiza la entrega de los paquetes ni previene el envío de paquetes duplicados. Estas últimas carencias, se encuentran disponibles en protocolo de transporte TCP. Dado que las aplicaciones WebRTC pueden ser muy variadas, algunos desarrolladores quieren optar por el rendimiento de UDP mientras que otros necesitarán la confiabilidad de TCP. Por ese motivo, surge entonces el protocolo SCTP, quien hace de intermediario entre las aplicaciones WebRTC y el protocolo UDP. Las características que ofrece este protocolo son:

- Reconocimiento de transferencia de datos, sin duplicados o errores
- Fragmentación de datos para adaptarse al MTU
- Entrega secuenciada de mensajes dentro de múltiples secuencias
- Tolerancia a fallos a nivel de red mediante el uso de *multi-homing* (conexión a múltiples redes)

El diseño de SCTP evita apropiadamente la congestión y ataques del tipo *flood* (sobrecarga de conexiones) y *masquerade* (falsa identidad).

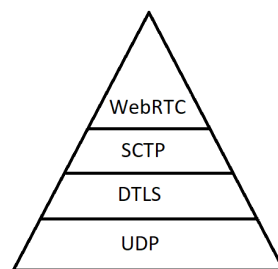


Figura A.6: Pila de protocolos en una aplicación WebRTC

Cabe mencionar que la comunicación entre SCTP y UDP no es directa, sino que entre ambos protocolos existe un protocolo llamado DTLS (*Datagram Transport Layer Security*) el cual provee las mismas garantías de seguridad ante ataques que el protocolo TLS (ej: escuchas, modificaciones, falsificaciones), pero para comunicaciones UDP.

Anexo B

Encuesta a los participantes de la sesión de pruebas

B.1. Acerca de Usted

- Nombre y Apellido
- Edad
- Ocupación
- Indique cuáles de las siguientes herramientas de comunicación de propósito general ha utilizado (Whats App, Slack, Skype, Google Hangouts)
- Indique si conoce alguna de estas herramientas que fueron utilizadas como inspiración para crear InfoMeeting (Apache Open Meetings, Cisco WebEx, Big Blue Button)

B.2. Acerca de Infomeeting

- ¿Qué fue lo que más le gustó de InfoMeeting?
- ¿Qué fue lo que menos le gustó InfoMeeting?
- ¿Qué funciones le parecieron más útiles? (Chat General, Chat Individual, El editor de código, Las Consultas, Pedir la palabra)

- ¿Qué tan intuitiva le pareció la interfaz? Poco intuitiva (1) - Muy intuitiva (5)
- ¿Qué tan útil le pareció el editor de código? Poco útil (1) - Muy útil (5)
- ¿Considera que InfoMeeting puede ser un complemento apropiado para la comunicación entre alumnos y docentes durante el dictado de un curso? (Sí, No, Tal vez)
- ¿Qué funcionalidad le agregaría a la herramienta?

B.3. Acerca de la interfaz y la experiencia de usuario

- ¿Qué tan intuitiva le resultó la interfaz (estructura, organización, etc)? Poco intuitiva (1) - Muy intuitiva (5)
- ¿Qué tan intuitivos le resultaron los íconos? Poco intuitivos (1) - Muy intuitivos (5)
- ¿Considera que las fuentes utilizadas, así como el tamaño de las mismas es adecuada? Poco adecuadas (1) - Muy adecuadas (5)
- ¿Considera que los colores utilizados tienen un buen contraste y favorecen la legibilidad? Muy en desacuerdo (1) - Muy de acuerdo (5)
- ¿Considera que las notificaciones sonoras al recibir mensajes son apropiadas? Muy en desacuerdo (1) - Muy de acuerdo (5)
- ¿Qué aspectos de la interfaz cree que pueden mejorarse?

B.4. Acerca de la performance

- ¿Qué dispositivo utilizó? (Notebook, PC de escritorio, Teléfono, Tablet)
- ¿Qué Sistema Operativo y navegador utilizó?
- ¿Que tipo conexión a internet utilizó? (Conexión por cable (cable coaxial), Conexión por línea telefónica (ADSL), Conexión Móvil 4G, Conexión Móvil 3G, Conexión por cable (fibra óptica), Otra)
- ¿Cómo le resultó el tiempo de carga y de respuesta de la aplicación? Muy lento (1) - Muy rápido (5)
- ¿Cómo notó la calidad del audio? Muy baja (1) - Muy alta (5)
- ¿Cómo notó la calidad del video? Muy baja (1) - Muy alta (5)

B.5. Preguntas sólo para el tutor

- ¿Cuán fácil le resultó llevar a cabo la sesión? Muy difícil (1) - Muy fácil (5)
- ¿Qué tan útiles encontró las notificaciones? Poco útiles o escasas (1) - Muy útiles (5)
- ¿Qué tan fácil le resultó darle la palabra a otro usuario? Muy difícil (1) - Muy fácil (5)
- ¿Qué funciones le agregaría a la herramienta para facilitar o potenciar la moderación de las sesiones?

Anexo C

Guiones utilizados durante las pruebas

C.1. Pasos para el tutor

C.1.1. Conexión: pruebas de audio y video

- Esperar a que se conecten todos los participantes, si los participantes no aparecen, intente recargar la aplicación.
- Preguntar si alguien tiene problemas de audio/video. Hacer esta misma pregunta por escrito en el chat general.
- Presentarse con los otros participantes: con nombre apellido y apellido. Conocer a los demás participantes y probar la funcionalidad de audio/video.

C.1.2. Dar la palabra

- Pregunte a los participantes si alguien quiere tener la palabra.
- Dar el control al participante que quiera tener la palabra durante aproximadamente 1 minuto. Recuperar el control o esperar que el participante lo devuelva.

C.1.3. Consultas y prueba de chat individual

- Notificar a los participantes que durante los próximos 3 minutos se pueden realizar y responder consultas libremente, usando la herramienta de consultas. Usted podrá ver los resultados de las consultas a medida que sean respondidas.
- Mientras los participantes prueban la herramienta de consultas, elija al menos dos participantes al azar y pruebe el chat individual.

C.1.4. Modo editor de código

- Pasar a modo editor de código
- Escriba código JavaScript (puede copiar que se encuentra a continuación) para que los participante puedan ver en tiempo real como usted escribe código.

C.1.5. Uso libre de la herramienta, prueba de bloquear participación

- Cuando termine de escribir el código, comunique a los participantes a partir de ahora pueden usar la herramienta de modo libre durante aproximadamente 5 minutos. Durante este tiempo otorgue la palabra a los participantes que se lo pidan.
- Elija un participante al azar y bloquee su participación, con el fin de probar esta característica.
- Cuando hayan transcurrido los 5 minutos, despida a los participantes y finalmente cierre la sesión.

C.2. Pasos para el resto de los participantes

C.2.1. Conexión: pruebas de audio y video

- Esperar a que se conecten los otros participantes, si los participantes no aparecen, intente recargar la aplicación.
- En caso de tener problemas con el audio o el video, utilice el chat general para notificar a los demás participantes de su problema.

C.2.2. Dar la palabra

- En un determinado momento, el moderador le preguntará a los participantes si quieren tener la palabra. Si lo desea, usted puede aceptar y hablar durante un minuto.

C.2.3. Consultas y prueba de chat individual

- Cuando usted (o el participante al que se le dio la palabra) termine, comience a crear y responder consultas. Usted podrá ver los resultados de las consultas a medida que éstas sean respondidas.
- Mientras prueba la herramienta de consultas, inicie un chat privado con algún participante al azar.

C.2.4. Modo editor de código

- Observe como el moderador utiliza el modo editor de código.

C.2.5. Uso libre de la herramienta

- Cuando el moderador lo indique, use la herramienta de modo libre durante aproximadamente 5 minutos. Si el moderador bloquea su participación durante este tiempo, no se preocupe, es parte de la prueba.
- Cuando hayan transcurrido los 5 minutos (si no fue bloqueado), despídase de los demás participantes y cierre la sesión.

Bibliografía

- [1] Ajjan, H., & Hartshorne, R. (2008). Investigating faculty decisions to adopt Web 2.0 technologies: Theory and empirical tests. *The internet and higher education*, 11(2), 71-80.
- [2] Al-Nuaim, H. A. (2012). The use of virtual classrooms in E-learning: A case study in king abdulaziz university, saudi arabia. *E-Learning and Digital Media*, 9(2), 211-222.
- [3] Al-Senaidi, S., Lin, L., & Poirot, J. (2009). Barriers to adopting technology for teaching and learning in Oman. *Computers & Education*, 53(3), 575-590.
- [4] Berge, Z. L. (1995). Facilitating computer conferencing: Recommendations from the field. *Educational Technology-Saddle Brook NJ-*, 35, 22-22.
- [5] Bertogna, M. L., Castillo, R. D., Soto, H., & Cecchi, L. (2007). Clases Sincrónicas Virtuales en la Enseñanza a Distancia: una implementación a bajo costo. En II Congreso de Tecnología en Educación y Educación en Tecnología.
- [6] Buckingham Shum, S., Marshall, S., Brier, J., & Evans, T. (2002). Lyceum: Internet voice groupware for distance learning.
- [7] Cabero Almenara, J., & Llorente Cejudo, M. C. (2007). La interacción en el aprendizaje en red: uso de herramientas, elementos de análisis y posibilidades educativas. *RIED. Revista iberoamericana de educación a distancia*, 10(2).
- [8] Caldeiro, G. P. (2014). La comunicación en línea y el trabajo colaborativo mediado por tecnologías digitales. *Austral Comunicación*, 3(1), 13-30.
- [9] De Suárez, M. P. O., Pineda-Báez, C., & Rees, G. P. (2016). Herramientas de comunicación y presentación de contenidos en pregrados virtuales colombianos. *Magis*, 8(17), 65.
- [10] Ghazal, S., Samsudin, Z., & Aldowah, H. (2015). Students' perception of synchronous courses using skype-based video conferencing. *Indian Journal of Science and Technology*, 8(30).

- [11] Ghirardini, B. (2011). E-learning methodologies: a guide for designing and developing e-learning courses. Food and Agriculture Organization of the United Nations.
- [12] Hyder, K., Kwinn, A., Miazga, R., & Murray, M. (2007). Synchronous e-learning. The eLearning Guild.
- [13] Itmazi, J. A., Megías, M. G., Paderewski, P., & Vela, F. L. G. (2005). A comparison and evaluation of open source learning management systems. *IADIS AC*, 1(11).
- [14] Jennings, R. B., Nahum, E. M., Olshefski, D. P., Saha, D., Shae, Z. Y., & Waters, C. (2006). A study of internet instant messaging and chat protocols. *IEEE Network*, 20(4), 16-21.
- [15] Johnson, G. M. (2006). Synchronous and asynchronous text-based CMC in educational contexts: A review of recent research. *TechTrends*, 50(4), 46.
- [16] Martin, F., Parker, M., & Allred, B. (2013). A Case Study on the Adoption and Use of Synchronous Virtual Classrooms. *Electronic journal of E-learning*, 11(2), 124-138.
- [17] Mason, R., & Romiszowski, A. (2003). *Computer-mediated Communication*.
- [18] Millán, I. F., Flores, S. R., Díaz, M., Conturzo, G., Aciar, S., & Baldiris, S. (2015). Herramienta tecnológica para la interacción sincrónica: moderador de chat automático. In *X Congreso sobre Tecnología en Educación & Educación en Tecnología (TE & ET)(Corrientes, 2015)*.
- [19] Park, Y. J., & Bonk, C. J. (2007). Synchronous learning experiences: Distance and residential learners' perspectives in a blended graduate course. *Journal of Interactive Online Learning*, 6(3), 245-264.
- [20] Park, Y. J., & Bonk, C. J. (2007). Is online life a breeze? A case study for promoting synchronous learning in a blended graduate course. *Journal of Online Learning and Teaching*, 3(3), 307-323.
- [21] Postel, J. (1980). User datagram protocol (No. RFC 768).
- [22] Rescorla, E., & Modadugu, N. (2012). *Datagram transport layer security version 1.2*.
- [23] Rogers, E. M. (2010). *Diffusion of innovations*. Simon and Schuster.
- [24] Serce, F. C., & Yildirim, S. (2006). A Web-Based synchronous collaborative review tool: A case study of an on-line graduate course. *Journal of Educational Technology & Society*, 9(2).
- [25] Sergiienko, A. (2015). *WebRTC Cookbook*. Packt Publishing Ltd.

- [26] Shoemaker, K., & van Stam, G. (2010, April). e-Piano, a case of music education via e-learning in rural Zambia. In Web Science Conference 2010.
- [27] Stewart, R. (2007). Stream control transmission protocol.
- [28] Tancredi, B. (2006). La comunicación síncrona en e-learning: cómo gestionar el chat. In Prácticas de e-learning (pp. 1000-1026). Editorial Octaedro.
- [29] VALVERDE B, J. (2002). Herramientas de comunicación síncrona y asincrónica.
- [30] Van Lokven, T. (2011). Review and Comparison of Instant Messaging Protocols. Computer Science Bachelor Thesis, 438006.
- [31] Wang, Y., & Chen, N. S. (2007). Online synchronous language learning: SLMS over the Internet. Innovate: Journal of Online Education, 3(3).
- [32] Wang, V., Salim, F., & Moskovits, P. (2013). Introduction to HTML5 WebSocket. In The Definitive Guide to HTML5 WebSocket (pp. 1-12). Apress, Berkeley, CA.
- [33] Zheng, L., & Rupp, D. S. (2005). Instant messaging: Architectures and Concepts. Institute of communication networks and computer engg, university of Stuttgart.