

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UN SISTEMA PROTOTIPO DE VIDEOCONFERENCIAS MEDIANTE WEBRTC PARA UN INSTITUTO TECNOLÓGICO

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN**

Christian Giovanni Basantes Guerra

DIRECTOR: ING. XAVIER CALDERÓN HINOJOSA

Quito, marzo 2023

AVAL

Certifico que el presente trabajo fue desarrollado por Christian Giovanni Basantes Guerra, bajo mi supervisión.



**M.Sc. XAVIER CALDERÓN HINOJOSA
DIRECTOR DEL TRABAJO DE TITULACIÓN**

DECLARACIÓN DE AUTORÍA

Yo Christian Giovanni Basantes Guerra, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.



CHRISTIAN GIOVANNY BASANTES GUERRA

DEDICATORIA

El presente trabajo se lo dedico principalmente a mis padres, que han sido mi inspiración toda mi vida, siempre me han enseñado buenos valores y me han dado todo lo que necesito para poder crecer como persona, los acontecimientos en la vida de mis padres me han demostrado que con mucho esfuerzo y dedicación puedo lograr lo que me proponga.

A toda mi familia y amigos que me han ayudado cuando más lo he necesitado.

A mí, porque a pesar de todo lo logré.

Christian Giovanni Basantes Guerra

AGRADECIMIENTO

A Dios y mis padres por haberme dado el don de la vida y poder disfrutar lo bueno y malo de este mundo, a mis padres, profesores y a la universidad que han ayudado a culminar esta etapa en mi vida de preparación profesional.

A mis amigos con los que he compartido todos estos años que siempre me apoyaron en situaciones buenas y malas.

Christian Giovanni Basantes Guerra

ÍNDICE DE CONTENIDO

AVAL.....	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO.....	V
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS	X
RESUMEN	XI
ABSTRACT.....	XII
1 INTRODUCCIÓN	1
1.1 OBJETIVOS.....	2
1.2 ALCANCE	2
1.3 MARCO TEÓRICO.....	5
1.3.1 COMUNICACIONES BASADAS EN WEB	5
1.3.2 APLICACIÓN WEB	6
1.3.3 WEBRTC	7
1.3.1 ARQUITECTURA WEBRTC.....	7
1.3.2 LA TECNOLOGÍA RTC EN LOS NAVEGADORES WEB.....	9
1.3.3 PRINCIPALES APIs EN JAVA SCRIPT	9
1.3.4 CÓDECS CON LOS CUALES FUNCIONA WEBRTC EN NAVEGADORES WEB TRADICIONALES.....	11
1.3.5 SERVIDORES XMPP.....	14
1.3.6 NODE. JS	15
1.3.7 REACT NATIVE	16
1.3.8 JAVASCRIPT	¡Error! Marcador no definido.
1.3.9 SAAS (Syntactically Awesome Style Sheets - hojas de estilo en cascada)	¡Error! Marcador no definido.
1.3.10 METODOLOGÍA KANBAN.....	18

2	METODOLOGÍA	20
2.1	DESARROLLO	20
2.1.1	ELABORACIÓN DE TABLERO KANBAN	20
2.1.2	REQUERIMIENTOS.....	21
2.1.3	ROLES Y PERMISOS	23
2.1.4	CASOS DE USO	24
2.1.5	INTERFAZ GRÁFICA.....	25
2.1.6	LÓGICA DE PROCESOS.....	28
2.2	IMPLEMENTACIÓN DEL SISTEMA.....	42
2.2.1	APLICACIÓN WEB	42
2.2.2	BASE DE DATOS	43
2.3	PRUEBAS PRELIMINARES.....	48
3	RESULTADOS Y DISCUSIÓN	56
3.1	AUTENTICACIÓN.....	57
3.2	SERVICIO DE MENSAJERÍA INSTANTÁNEA (CHAT).....	60
3.3	COMPARTICIÓN D EPANTALLA	60
3.4	GRABACIÓN DE REUNIÓN	61
3.5	TRANSMISIÓN EN YOUTUBE	62
3.6	ASIGNACIÓN DE SEGURIDAD A LA SESIÓN.....	64
3.7	NOTIFICACIONES AL ADMINISTRADOR.....	65
3.8	PÁGINA DE AYUDA	66
3.9	PRUEBAS EN EQUIPOS MÓVILES	67
4	CONCLUSIONES Y RECOMENDACIONES.....	68
4.1.	CONCLUSIONES.....	68
4.2.	RECOMENDACIONES	69
5	REFERENCIAS BIBLIOGRÁFICAS	73
	ANEXOS.....	75
	ANEXO A.....	76
1	ENCUESTA REALIZADA PARA DETERMINAR LOS REQUERIMIENTOS DEL SISTEMA DE VIDEOCONFERENCIA.....	76
2	ENCUESTA REALIZADA PARA DETERMINAR LA SATISFACCIÓN DEL USO DEL PROTOTIPO.	80

3	MANUAL DE USUARIO PARA CREAR SALAS DE VIDEOCONFERENCIA.....	83
4	MANUAL DE ADMINISTRADOR PARA LA ADMINISTRACIÓN DE MODERADORES.....	95
5	PRUEBAS ADICIONALES EN COMPUTADORES.....	100

ÍNDICE DE FIGURAS

Figura 1.1. Arquitectura del sistema de videoconferencias y chat mediante WebRTC	5
Figura 1.2. Modo de captación de datos de una aplicación Web	6
Figura 1.3. Arquitectura WebRTC	7
Figura 1.4. Modelo Trapezoidal WebRTC	9
Figura 1.5. Navegadores Web desktop que soportan WebRTC	13
Figura 1.6. Navegadores Web móvil que soportan WebRTC	13
Figura 1.7. Arquitectura de XMPP cliente servidor descentralizado	15
Figura 2.1. Tablero KANBAN de tareas a realizar para el desarrollo del sistema.	21
Figura 2.2. Diagrama de casos de uso de Administrador General.....	24
Figura 2.3. Diagrama de casos de uso de Usuario/Moderador.....	25
Figura 2.4. Sketch de pantalla de bienvenida.....	26
Figura 2.5. Sketch de confirmación para inicio de sesión.....	26
Figura 2.6: Sketch de pantalla de ingreso de credenciales del usuario.	27
Figura 2.7: Sketch de pantalla de opciones en la sesión.....	27
Figura 2.8. Diagrama de actividades del proceso de registro y autenticación de usuarios.	29
Figura 2.9. Diagrama de secuencia del proceso registro de usuario.	30
Figura 2.10. Diagrama de secuencia del proceso inicio de sesión	31
Figura 2.11. Diagrama de actividades del proceso chat.	33
Figura 2.12. Diagrama de secuencia del proceso chat.....	34
Figura 2.13. Diagrama de actividades del proceso para el módulo perfil.....	35

Figura 2.14. Diagrama de secuencia del proceso edición de perfil.....	37
Figura 2.15. Diagrama de actividades del proceso notificación.....	38
Figura 2.16. Diagrama de secuencia del proceso notificación.....	38
Figura 2.17. Diagrama de actividades para la creación de sesión.....	39
Figura 2.18. Diagrama de secuencia para la creación de sesión.....	40
Figura 2.19. Diagrama de clases.....	41
Figura 2.10. Pantalla de bienvenida tomada en las pruebas preliminares del sistema.....	43
Figura 2.21. Pantalla de máquina virtual Ubuntu 20.0 instalada en VMWare ESXi para el servidor de videoconferencia.....	44
Figura 2.22 Ingreso al sistema virtualizado en UBUNTU mediante servicio SSH, VMWare en servidor real del ISTS.....	45
Figura 2.23. Pantalla MySQL para ingreso a la base de datos del sistema de videoconferencia.....	45
Figura 2.24. Servidores activos para el correcto funcionamiento del sistema de videoconferencia.....	46
Figura 2.25. Diagrama modelo relacional.....	47
Figura 2.26. Contenido de la tabla prosody de la base de datos db_usuarios_reuniones.....	48
Figura 2.27. Asignación de IP pública al servidor local y apertura de puertos.....	48
Figura 2.28. Tablero KANBAN actualizado.....	49
Figura 2.29. Código de programación para autenticación de usuario en el sistema.....	51
Figura 2.30. Programación de renderización de la página, ingreso de usuario y contraseña del usuario.....	52
Figura 2.31. Programación de <i>Props</i> (a) y <i>State</i> (b).....	53
Figura 2.32. Programación de métodos a utilizar.....	54

Figura 3.1. Tablero KANBAN actualizado.	56
Figura 3.2. Creación de Sala conferencia, para ejemplo "Digitales 1".	57
Figura 3.3. Formulario de ingreso para moderador con usuario y clave asignada.	58
Figura 3.4. Error en la autenticación del moderador.....	58
Figura 3.5. Salón virtual "Auditorio 1" activo en modo moderador.	59
Figura 3.6. Ingreso de participantes a la sesión activa.	59
Figura 3.7. Reunión con cuatro (4) participantes y envío de mensajes mediante el sistema de chat de la prototipo.	60
Figura 3.8. Activación de modo compartición de pantalla.....	61
Figura 3.9. Presentación de pantalla compartida.	61
Figura 3.10. Activación de grabación de sesión.	62
Figura 3.11. Grabación de Sesión, ejecutada por el moderador.....	62
Figura 3.12. Solicitud de transmisión por YouTube.	63
Figura 3.13. Activación de transmisión con el código de transmisión obtenido desde YouTube.....	63
Figura 3.14. Transmisión por YouTube activado. se presenta la notificación en la parte superior junto al nombre de la sala.	64
Figura 3.15. Transmisión de la sesión en la prototipo YouTube.	64
Figura 3.16. Configuración de seguridad para ingreso a la sesión "Auditorio 1"...	65
Figura 3.17. Ventana auxiliar de envío de notificaciones.....	66
Figura 3.18. Sub menú del sistema para ingreso al módulo de ayuda de la prototipo.....	66
Figura 3.19. Módulo de ayuda con la explicación del funcionamiento del sistema.	67
Figura 3.20. prueba realizada en smartphone con Android OS 12.	67

ÍNDICE DE TABLAS

Tabla 2.1. Especificaciones técnicas de máquinas usadas para la virtualización.	42
Tabla 2.2: Especificaciones técnicas de equipos usados e implementados para virtualización de servidor de almacenamiento dentro del ISTS.....	43
Tabla 2.3. Atributos tabla prosody.....	46
Tabla 2.4. Puertos abiertos en el servidor de videoconferencia.....	48

RESUMEN

Con la llegada de las restricciones de movilidad, debido a la emergencia sanitaria suscitada alrededor del mundo, por la pandemia de COVID-19 era necesario mantener un canal de comunicación que permita la interacción de las personas para continuar con las actividades cotidianas. Es en este momento donde las plataformas digitales cobran mayor relevancia y entran en auge, desarrollando y mejorando las aplicaciones existentes sobre todo en el ámbito comercial y educativo. Para este último, se desarrollaron prototipos pagados que permiten ingresar a un salón virtual, como Zoom, Microsoft Teams, entre otras, donde los estudiantes y docentes pueden simular el ambiente educativo en un aula de clases convencional. Debido a esto, se ha desarrollado un prototipo de aplicación web que permite realizar las comunicaciones del Instituto Superior Tecnológico Sucre mediante videoconferencia utilizando tecnología WebRTC. El presente trabajo de titulación consta de cuatro capítulos que se mencionan a continuación.

En el capítulo uno el lector podrá encontrar información relevante sobre el proyecto, como objetivos, alcance y el marco teórico que contiene referencias sobre las comunicaciones web mediante el uso de tecnología WebRTC, el tipo de arquitectura del sistema, los APIs más comunes soportados en sistemas de videoconferencias. Posterior a esto la descripción del funcionamiento de los servidores utilizados para la implementación del prototipo. En el capítulo dos se presenta la metodología utilizada para el desarrollo del aplicativo Web. Misma que se encuentra basada en Kanban, señalando los requerimientos funcionales y no funcionales, el diseño y la implementación del sistema.

Dentro del capítulo tres se presentan las pruebas realizadas en el entorno de trabajo para la evaluación del funcionamiento del sistema, una vez realizadas las pruebas se ejecuta un análisis de estas y se corrigen posibles errores. Finalmente, en el capítulo cuatro se realiza una reseña con las conclusiones y recomendaciones del estudiante referente al prototipo propuesto e implementado.

PALABRAS CLAVE: Videoconferencia, Instituto Superior Tecnológico Sucre, WebRTC, Comunicación en tiempo real, Web Real-Time Communications.

ABSTRACT

With the arrival of mobility restrictions, due to the health emergency around the world, by the pandemic of COVID-19, it was necessary to maintain a communication channel that allows the interaction of people to continue with daily activities. It is at this moment where digital platforms become more relevant and are booming, developing and improving existing applications, especially in the commercial and educational fields. For the latter, paid platforms have been developed that allow access to a virtual classroom, such as Zoom, Microsoft Teams, among others, where students and teachers can simulate the educational environment in a conventional classroom. Therefore, a prototype of a web application has been developed that allows the “Instituto Superior Tecnológico Sucre” to communicate through videoconferencing using WebRTC technology. This degree work consists of four chapters which are mentioned below.

In chapter one the reader will find relevant information about the project, such as objectives, scope and the theoretical framework that contains references about web communications through the use of WebRTC technology, the type of system architecture, the most common APIs supported in videoconferencing systems. This is followed by a description of the servers used for the implementation of the prototype. In chapter two the methodology used for the development of the Web application is presented. This is done in Kanban, mentioning the functional and non-functional requirements, the design and implementation of the system.

Chapter three presents the tests performed in the work environment for the evaluation of the system's operation, which at the end are analyzed and possible errors are corrected. Finally, in chapter four there is a review with the conclusions and recommendations of the researcher regarding the proposed and implemented prototype.

KEYWORDS: Videoconference, Technological Institute Sucre, WebRTC, Real time communication, Web Real-Time Communications.

1. INTRODUCCIÓN

Actualmente el mundo vive un avance tecnológico que se desarrolla de manera acelerada lo que ha permitido que las personas progresen y adquieran los conocimientos necesarios para lograr cubrir necesidades y alcanzar una mejor calidad de vida. Estas tecnologías convergen en el área de la informática y las comunicaciones o transferencias de información, por lo que se ha denominado la “era de la informática y la información” [2].

Últimamente leer un periódico, ver las noticias, las comunicaciones del trabajo, o separar entradas para las diferentes actividades de ocio es mucho más rápido y sencillo gracias a que el internet, a día de hoy, se encuentra presente en la mayoría de los dispositivos electrónicos que se utilizan a diario como smartphones, computadores portátiles, y cualquier dispositivo que tenga conectividad a la red de datos. Lo que demuestra un verdadero avance tecnológico, mismo que han cambiado las costumbres y modos de vida.

De igual manera para que lo descrito anteriormente sea posible, ha sido necesario la evolución de las páginas web, páginas que en función de las necesidades y exigencias de los usuarios se vieron en la obligación de realizar cambios y actualizaciones en las tecnologías de desarrollo como HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), JavaScript, entre otros, con la finalidad de cubrir los requerimientos siendo más funcionales y completas.

Debido a la situación de emergencia sanitaria que se vive actualmente en todo el mundo por la pandemia de COVID-19, se pudo observar un incremento de uso de las redes sociales y plataformas como Microsoft Teams o Zoom. En el ámbito educacional las plataformas tienen un costo para ofrecer su máxima capacidad y opciones.

La ejecución de este proyecto tiene como finalidad el desarrollo de un sistema de videoconferencia con base en los protocolos WebRTC de código abierto, lo que permite al usuario realizar llamadas, conferencias e intercambio de datos de una forma eficaz y sencilla con la ventaja de que no deberá desembolsar valores

económicos por concepto de licencia de uso para tener el máximo rendimiento y opciones del prototipo implementada.

1.1 OBJETIVOS

El objetivo general de este Proyecto Técnico es:

Desarrollar un sistema prototipo de videoconferencias mediante el uso de WebRTC para el Instituto Superior Tecnológico Sucre en la ciudad de Quito.

Los objetivos específicos del Proyecto Técnico son:

- Analizar los fundamentos teóricos para la implementación de un sistema de videoconferencias y chat mediante WebRTC.
- Diseñar el prototipo especificado en función de sus módulos.
- Implementar los módulos descritos en el alcance del sistema prototipo.
- Validar el prototipo en base a pruebas funcionales.

1.2 ALCANCE

Para el desarrollo de este prototipo se tomarán requerimientos y se los determinará a partir de 3 reuniones conjuntas con miembros del Consejo de Dirección del Instituto Superior Tecnológico Sucre, Administradores de los servidores y especialistas del área de la Informática.

Su implementación se llevará a cabo con la virtualización de los servidores necesarios para la implementación de WebRTC en un equipo del Instituto y se utilizará Ubuntu server, distribución basada en Debian, donde se instalará un servidor web para ofrecer una interfaz al usuario, que facilita la realización de las videoconferencias, tanto en la red interna como externa. Con su uso se pretende integrar gran cantidad de usuarios y establecer comunicación en tiempo real en navegadores web desde un dominio específico de la institución, que requiere la autenticación y no generará incompatibilidad por abarcar cuentas diferentes. Se tratará de una red de nodos que facilitará compartición de información y el servicio será personalizado a las necesidades de los docentes. Los datos, el audio y video se transmitirá entre pares orientado a navegadores Web creando un sistema de comunicación en tiempo real con múltiples usuarios, inicialmente dos salas con diez

usuarios participantes en cada una hasta llegar al máximo y buscar un equilibrio de uso de recursos en el servidor y evitar saturación del mismo.

Los usuarios registrados en la base de datos del Instituto Tecnológico tendrán roles con los cuales se dará acceso a determinados módulos. El sistema tendrá los siguientes roles:

Administrador general:

Será el encargado de monitorear el correcto funcionamiento del sistema a nivel de consola en el servidor y podrá realizar la creación, lectura o admisión, actualización, y eliminación, CRUD (Create, Read, Update, Delete), de los usuarios con roles.

Moderador conferencista:

Controlará la petición de videoconferencia, así como los privilegios de inicio, finalización, grabación y transmisión en vivo de la misma en el prototipo YouTube, deberá crear una contraseña la cual permitirá que los usuarios participantes puedan unirse a la videoconferencia.

Usuario participante:

Se puede unir a una videoconferencia mediante una dirección Web donde requiere de una contraseña que será proporcionada por el moderador conferencista.

Dentro del alcance del proyecto se pretende desarrollar un prototipo de aplicación web que tendrá los siguientes módulos:

Módulo de autenticación:

Permitirá la autenticación tanto a moderadores conferencistas como a usuarios participantes quienes estarán registrados en la base de datos del Instituto, los mismos que deberán acceder mediante las credenciales asignadas, nombre de usuario y contraseña, para así poder validar el ingreso en la interfaz web; una vez creada las salas de videoconferencias se podrá invitar a usuarios participantes a través de un enlace web y una contraseña generada por el moderador conferencista. El correcto desarrollo de este módulo es de suma importancia debido a que con ella se evita que cualquier usuario que conozca el nombre de dominio de la página web de videoconferencias haga uso del mismo.

Módulo de comunicación de usuarios:

Permitirá establecer una conexión entre usuarios mediante el uso de WebRTC en navegadores Web en ordenadores y dispositivos móviles.

Módulo de mensajería instantánea:

Facilitará la transferencia de mensajes entre los participantes, utilizando la librería Socket.io para comunicaciones en tiempo real entre dispositivos. Incluye los módulos de Real Time Client y Real Time Server.

Módulo de compartición de pantalla:

Permitirá a los usuarios (moderador conferencista y usuario participante) compartir la pantalla y audio ya sea la pantalla de todo el escritorio, pantalla del navegador web o solo una pestaña del navegador.

Módulo de grabación de videoconferencias:

Permitirá grabar las reuniones de video en formato mp4 y las guardará en el disco de almacenamiento del servidor mediante el servidor FFmpeg, que se encarga de grabarlas y convertirlas.

Módulo de transmisión en vivo en YouTube:

Permitirá transmitir en vivo a un usuario registrado en YouTube mediante una clave de transmisión proporcionada por YouTube y hará uso del servidor FFmpeg que permite hacer streaming de audio y video.

Módulo de ayuda:

Contará con un manual de uso del software donde mediante ilustraciones gráficas se mostrará cómo utilizar el prototipo de videoconferencias y sus herramientas de autenticación, envío de mensajes, compartición de pantalla, grabación de videoconferencias y transmisión en vivo en YouTube.

Módulo de notificaciones:

Permitirá conocer algún fallo durante el proceso videoconferencia o insatisfacciones del usuario (moderador conferencista o usuario participante).

A continuación, en la Figura 1.1, se muestra un diagrama de conexiones físicas de los distintos equipos con el servidor principal.

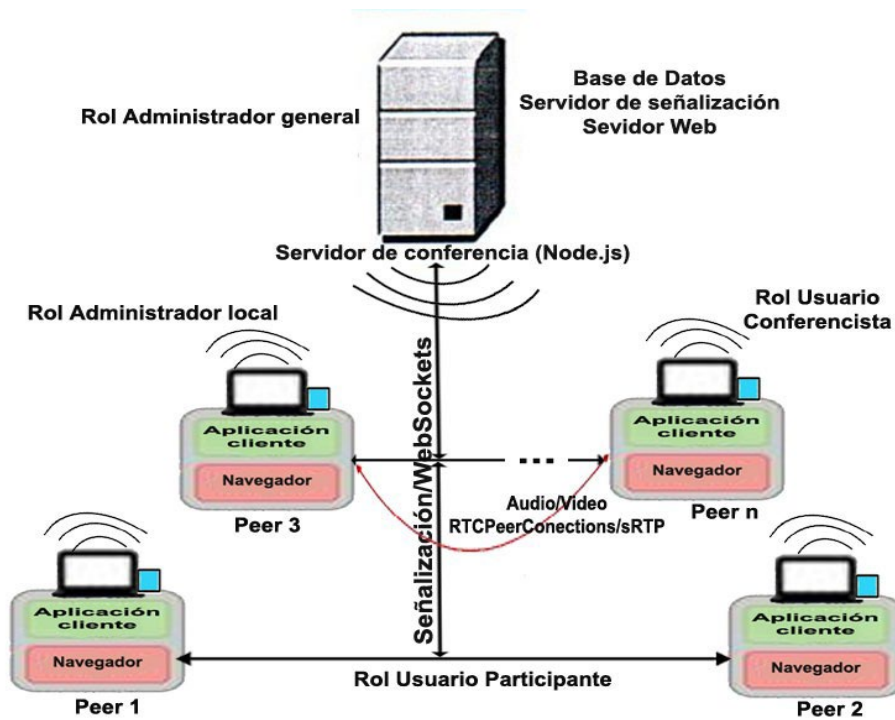


Figura 1.1. Arquitectura del sistema de videoconferencias y chat mediante WebRTC [1].

De la misma manera, se realizarán pruebas de integración, verificación de la consistencia de la información en la base de datos y pruebas de funcionalidad de los campos de cada formulario.

El sistema será evaluado inicialmente con dos salas de videoconferencias con un máximo de diez participantes en cada una de ellas, hasta llegar a la capacidad máxima siempre buscando un equilibrio del uso óptimo del recurso del servidor.

El desarrollo del prototipo de software tendrá su base en el marco de trabajo Kanban. Por lo tanto, el trabajo de titulación ofrece un producto final demostrable [1].

1.3 MARCO TEÓRICO

1.3.1 COMUNICACIONES BASADAS EN WEB

Cada vez son más las aplicaciones y prototipos que día a día aparecen brindando los diferentes servicios de establecimientos financieros, empresas privadas, sector

de salud y educación, este último es quien ha sobresalido en el uso de este tipo de tecnología pues han optado por aplicar el método enseñanza - aprendizaje mediante salones virtuales, lo que se traduce en el uso masivo de sistemas que emplean el protocolo WebRTC, Real Time Communication Web, para la transmisión e intercambio de información. Con esta transformación los navegadores Web deben estar siempre actualizados añadiendo varios servicios y funcionalidades a sus prototipos con el objetivo de permitir que estos puedan interactuar con otros navegadores y aplicaciones admitiendo la ejecución de estas en tiempo real [2].

1.3.2 APLICACIÓN WEB

Una aplicación Web son herramientas a las que el usuario puede acceder mediante un navegador como cliente de una aplicación o programa cuyos datos se encuentran almacenados en la nube (cloud) o en un servidor interno, permitiendo la interacción entre estos y el usuario; por tal razón su funcionamiento está basado en el modelo cliente-servidor. Una de las particularidades más importantes de este tipo de aplicaciones es que no es necesario la instalación de un software adicional en la máquina dónde se va a ejecutar este programa [3].

En la figura 1.2 se observa un diagrama de captación de datos de las aplicaciones Web.



Figura 1.2. Modo de captación de datos de una aplicación Web [3].

1.3.3 WEBRTC

WebRTC¹, también conocido como Protocolo Web de Tiempo Real es una tecnología que a lo largo del tiempo y con la nueva “normalidad” es considerada como una gran innovación debido a que permite a los desarrolladores de aplicaciones el uso de este para la creación de programas multimedia multifuncionales tales como videojuegos, sistemas de streaming, prototipos de educación y videoconferencias donde se permite el uso compartido de información, datos, imágenes, audio y video de manera inmediata. WebRTC es un protocolo Open Source y fue publicado por Google en el 2010, es importante mencionar que dentro de la WebRTC existe un sin número de componentes que abarcan códecs, motores de medios y capa de transporte [4].

1.3.4 ARQUITECTURA WEBRTC

La arquitectura de WebRTC, está compuesta por tres partes fundamentales que se detallan a continuación.

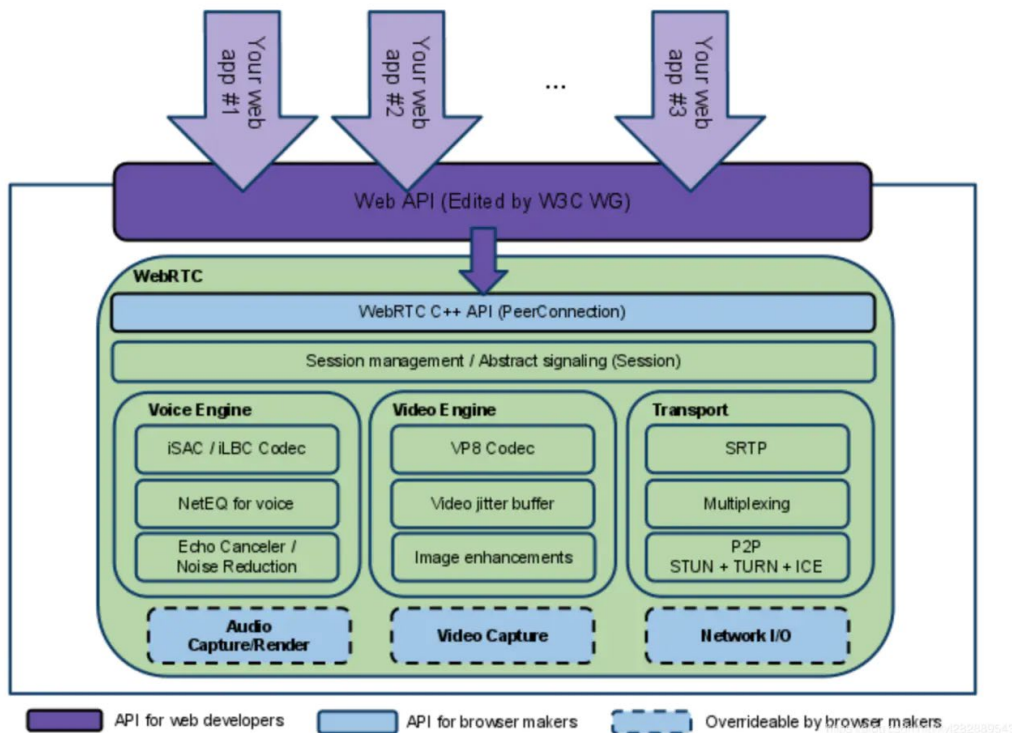


Figura 1.3. Arquitectura WebRTC [5].

¹WebRTC: *Web* Real-Time Communication

En la figura 1.3 se puede visualizar un esquema detallado referente a la arquitectura del protocolo WebRTC.

En los siguientes párrafos se detallará cada uno de los bloques observados en la figura anterior [6].

- API desarrolladores Web.

Un API² es una Interfaz de la Programación de Aplicaciones que se utilizan para crear e integrar aplicaciones con WebRTC. Este comúnmente es conocido como un paquete de software que permite interactuar con la aplicación en el ámbito de desarrollo siendo JavaScript el lenguaje necesario para ser implementado; y así obtener los recursos de la máquina, configurar la conexión entre pares, abrir un canal de transmisión de datos, obtener los candidatos, donde un candidato describe los protocolos y el enrutamiento necesarios para que WebRTC pueda comunicarse con un dispositivo remoto, entre otros.

- API para desarrolladores de navegadores.

Es una interfaz o API con la que los desarrolladores de navegadores Web permiten la conectividad entre pares utilizando la tecnología WebRTC.

- Control de sesiones.

Capa encargada de la gestión y control de sesión, permite la configuración de llamadas y la capa de administración. Esto quiere decir que es la encargada del protocolo de señalización que el desarrollador usará para la aplicación.

- Motor de voz, video.

Motor de voz, video es un API que sirve para la cadena de medios de audio y video respectivamente, esto desde el controlador madre de audio y video.

- Transporte Sesión.

Protocolos que usan como transporte de información para la transmisión de información en tiempo real.

Como se muestra en la figura 1.4 en este modelo de trabajo se tienen dos servidores web que están comunicados mediante un protocolo de señalización

² API: Application Programming Interfaces

pudiendo ser Jingle o SIP (Session Initiation Protocol), estos dos servidores están en constante sincronización y permiten presentar la aplicación a clientes que se encuentran asociados a diferentes servidores [7].

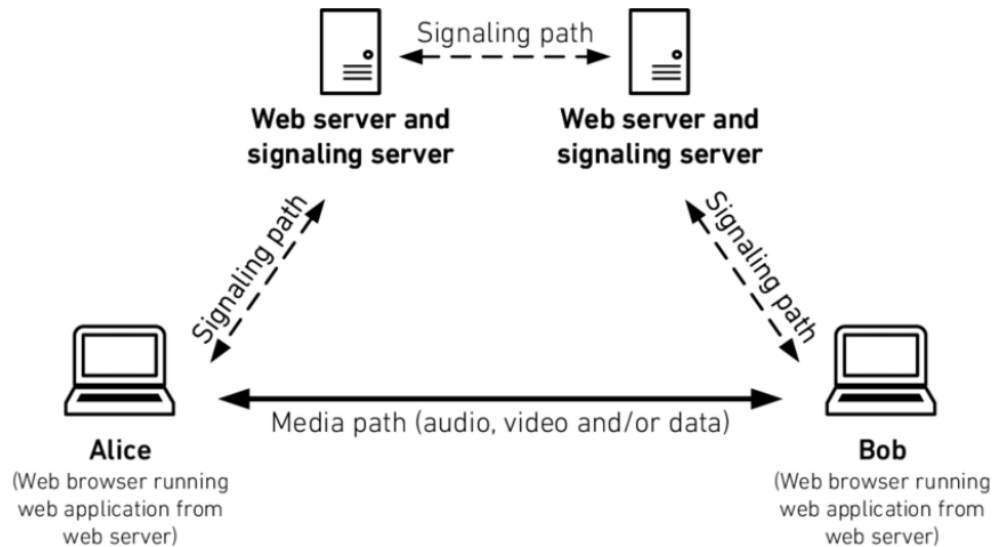


Figura 1.4. Modelo Trapezoidal WebRTC [6].

1.3.5 LA TECNOLOGÍA RTC EN LOS NAVEGADORES WEB

WebRTC es una tecnología que a lo largo y el transcurso del tiempo ha pasado de ser un recurso no previsto a ser hoy en día uno de los recursos más esenciales que las aplicaciones Web necesitan, por el mismo hecho de permitir aplicaciones multimedia, es decir, que se puede transmitir datos en tiempo real sin necesidad de instalar un cliente dentro del equipo donde se va a ser uso de este prototipo.

WebRTC revoluciona completamente la capa de transporte logrando su objetivo primordial que es la comunicación entre navegadores, los estándares no definen en si los requisitos que se necesita para este proceso de interoperabilidad [8].

1.3.6 PRINCIPALES APIs EN JAVA SCRIPT

EL API básicamente, ayuda al desarrollador a crear una interfaz por la que se permite implementar códigos de programación, que acceden a los recursos de los dispositivos; recursos como, cámara web y micrófonos periféricos que básicamente son necesarios para la comunicación multimedia en tiempo real. Para permitir el intercambio de paquetes de información WebRTC implementa la siguiente API:

1.1.1.1 RTC Peer Connection

Esta interfaz o API es fundamentalmente el que permite la comunicación entre equipos locales y remotos, así mismo es el responsable de que ambas partes puedan establecer, mantener y monitorear la conexión y no obstante también se pueda cerrar la misma. Para que esta comunicación sea funcional es necesario un proceso de intercambio de mensajes para establecer la comunicación y por ende los parámetros que prevalecerán de esta comunicación [8].

La interfaz puede tener listeners para estar a la espera como:

- **RTCPeerConnection.iceConnectionState:**
Esta es una de las numerosas funciones que tiene el API, su principal prestación es que devuelve el estado de la conexión ICE, teniendo estados o valores como: “new”, “checking”, “connected”, “completed”, “failed”, “disconnected”, “closed” [9].
- **RTCPeerConnection.iceGatheringState:**
Esta función como resultado entrega un valor predefinido del tipo `RTCIceGatheringState`, que describe el estado de la conexión, este valor puede ser: “new”, “gathering”, “complete” [9].
- **RTCPeerConnection.localDescription:**
Esta es otra de las funciones pertenecientes a WebRTC, lo que permite esta API, es la descripción de la sesión de la máquina local del cliente (`ClientEndPoint`). Esta información será reenviada al cliente remoto para su respectiva descripción [9].
- **RTCPeerConnection.peerIdentity:**
Una vez que se ha realizado la conexión exitosa entre los computadores local y remoto, esta conexión entre pares devuelve la identidad de la confirmación de la conexión en forma de dos valores: el nombre de dominio (`ipd`) y el nombre (`name`) [9].
- **RTCPeerConnection.remoteDescription:**
Este API da como resultado la descripción de sesión de la máquina remota (`ClientEndPoint`) de la conexión. Si no se realiza la conexión devuelve null [9].

- **RTCPeerConnection.signalingState:**
Este API es capaz de brindar información sobre el estado de la señal de la conexión local. También es la encargada de emitir la sesión de descripción (SDP) de la oferta, que define la configuración de las conexiones como el objeto `MediaStream` asociado [9].
- **RTC Data Channel:**
Este API brinda la facilidad de contar con un canal de datos bidireccional para la comunicación entre los computadores. Para hacer uso de esta API es necesario `RtcPeerConnection` ya que así se establece la conexión con un par [9].
- **Get User Media (MediaStream):**
La función principal de `Get User Media (MediaStream)` es acceder a los flujos de datos como la cámara y el micrófono del usuario, es importante mencionar que para esto se debe contar con los privilegios del usuario ya que se usará periféricos que están dentro del computador como dispositivos de entrada de audio, cámara, micrófono, y escritorio remoto [9].

1.3.7 CÓDECS CON LOS CUALES FUNCIONA WEBRTC EN NAVEGADORES WEB TRADICIONALES.

Siempre es primordial tener un lenguaje de comunicación y sincronización dentro del proceso de comunicación, con esto un códec, es un conjunto de algoritmos de compresión y descompresión, para archivos de voz o video, y que los mismos puedan ser transmitidos más rápido a través de Internet o almacenados localmente en un computador.

La API de WebRTC, hace posible la construcción de sitios web y aplicaciones que permiten a los usuarios comunicarse en tiempo real, utilizando audio y/o video, así como datos opcionales y otra información. Para comunicarse, los dos dispositivos deben poder acordar un códec de entendimiento mutuo para cada pista o fragmento de audio o video, de modo que puedan comunicarse y presentar correctamente los medios compartidos. A continuación, se describen los códecs que los navegadores deben implementar, así como otros códecs que algunos o todos los navegadores admiten para WebRTC. Entre los códecs usado para video y audio se encuentran:

- VP8 formato de compresión de video:

VP8 es un códec de video muy conocido que básicamente se utiliza para la compresión de video, este códec es de código abierto y fue creado por Google es muy confiable y con la capacidad de ocultar errores dentro de la transmisión de información; además cuenta con una alta compresión de video, y es compatible con códec como ACC (Advanced Audio Coding), MP3 o incluso dentro del contenedor MP4 [10].

Para audio:

- G.711:

G.711 es un códec utilizado principalmente en los servidores de telefonía y VoIP, el principal funcionamiento es que transforma las señales de voz analógicas a una salida digital de 64 Kbps [11].

- G.722:

Es un códec que con una frecuencia de muestreo de 16 KHz, obteniendo así una mejor calidad y claridad de audio. Este códec es capaz de operar en banda ancha de 48, 56 y 64 Kbps [11].

- iSAC:

Internet Spech Audio Códec, es un códec el cual es usado por varias aplicaciones de VoIP y para la transmisión en tiempo real, fue creado por Global IP Solution [9]. Este códec trabaja con velocidades de transmisión de 10 y 32 Kbps.

- iLBC:

Internet Low Bitrate Códec este códec es de tipo híbrido ya que es capaz de trabajar en dos frecuencias, ya que es capaz de trabajar con velocidades de transmisión de 13.33 y 15.2 Kbps, es de código abierto y es muy utilizado en aplicaciones que necesitan comunicación VoIP de carácter robusto y confiable [12].

Cada una de las plataformas en cuanto a navegadores cuentan por lo menos con un códec de audio y un códec de video con los que WebRTC es compatible. El navegador Google Chrome es el que más características, prestaciones y

funcionalidades tiene para adoptar la tecnología WebRTC. A continuación, se presenta una figura con los navegadores que soportan WebRTC.

IE	Edge *	Firefox	Chrome	Safari	Opera
	12-14	2-21	4-22		10-17
	³ 15-18	22-43 ⁻	23-55 ⁻	3.1-10.1	18-42 ⁻
6-10	79-81	44-76	56-81	11-13	43-67
11	83	77	83	13.1	68
		78-79	84-86	14-TP	

Figura 1.5. Navegadores Web desktop que soportan WebRTC

Fuente: [13].

En la figura 1.5 se observan los diferentes navegadores web presentes en aplicaciones de escritorio que soportan la integración de WebRTC para la transmisión de paquetes de información en tiempo real, cabe recalcar que a pesar de que el navegador web soporte esta tecnología; también es importante que el mismo soporte las características representativas de WebRTC que son: ingreso a una reunión, modo de presentación pantalla, modo de compartición de pantalla y control remoto. De la misma forma en la figura 1.6 se observan los diferentes navegadores web móviles que soportan esta tecnología. Además, se puede señalar que, en color rojo se encuentran marcadas las versiones de cada navegador que no son compatibles, y en color verde las que son compatibles con WebRTC.

iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser
3.2-10.3										
11-13.3		2.1-4.4.4	12-12.1				4-10.1 ⁻			
13.5	all	81	46	81	68	12.12	11.2 ⁻	10.4 ⁻	7.12 ⁻	2.5
14.0										

Figura 1.6. Navegadores Web móvil que soportan WebRTC

Fuente: [13].

1.3.8 SERVIDORES XMPP

El protocolo extensible de mensajería y comunicación de presencia (XMPP - The Extensible Messaging and Presence Protocol) es un protocolo de código abierto principalmente pensado para comunicaciones en tiempo real, este protocolo hace uso de XML para el intercambio de información, y básicamente el funcionamiento está orientado al envío de pequeños mensajes XML desde una entidad hacia otra entidad en tiempo real, lo que en la era actual se llamaría mensajes push o chat.

El protocolo XMPP y las extensiones de este servicio obligatoriamente deben cumplir y añadir funcionalidades que permitan que la transferencia de mensajes sea segura, estas funcionalidades agregan algoritmos de autenticación, control de acceso y privacidad. Una de las características muy útiles, es que el protocolo XMPP tiene la capacidad de involucrarse en problemas de internacionalización, localización y puede trabajar en conjunto con los modelos de presencia y mensajes instantáneos RFC2778 y RFC2779, logrando así sala de chatrooms [14].

1.1.1.2 ARQUITECTURA XMPP

Siempre es necesario conocer la arquitectura y en este caso el protocolo XMPP, se maneja bajo la arquitectura cliente-servidor descentralizada, muy similar a la que se maneja en el protocolo WWW o muy similar al servicio de Exchange o correo electrónico. Lo interesante de trabajar con una arquitectura cliente-servidor descentralizada es que esta permite una separación inteligente de responsabilidades, es decir, los desarrolladores de clientes pueden centrarse en la experiencia del usuario y los desarrolladores de servidores pueden centrarse en la fiabilidad y la escalabilidad, es mucho más fácil para las organizaciones gestionar frente a una tecnología propuesta bajo la arquitectura peer-to-peer. XMPP es bastante robusta, figura 1.7 [14].

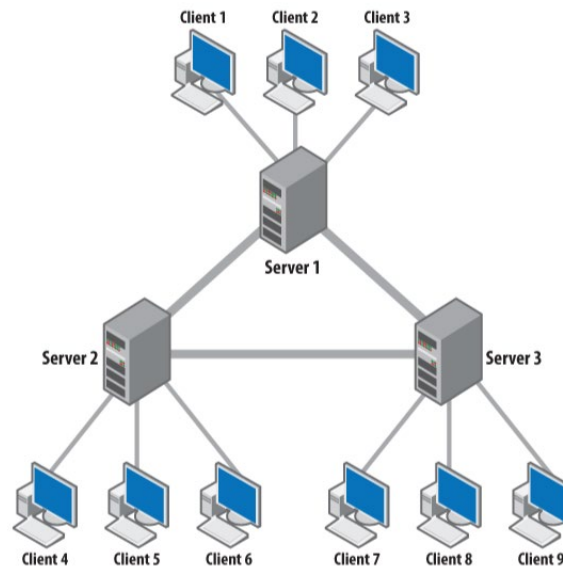


Figura 1.7. Arquitectura de XMPP cliente servidor descentralizado

Fuente: [14]

1.3.9 NODE. JS

Node JS es un entorno de ejecución de JavaScript controlado por eventos y está principalmente pensado para crear aplicaciones escalables, Node JS está construido sobre el motor JavaScript V8 de Google Chrome y gracias a esto presenta varias características tales como: su biblioteca funciona de manera rápida y permite establecer múltiples conexiones al mismo tiempo evitando bloqueos.

La forma de funcionamiento de esta API es que la misma ofrece un bucle de eventos, otra de las características importantes es que simplifica la comunicación entre otras que han hecho que Node.JS aporte con un gran cambio en la tecnología de transmisión de información, además, es muy popular y confiable [15].

Entre las ventajas de node JS están [16]:

- El motor JavaScript V8 utilizado en Google Chrome es rápido y puede ejecutar miles de instrucciones por segundo.
- Node.js fomenta un estilo de codificación asíncronica, un código más rápido para administrar la concurrencia y evitar subprocessos múltiples que podrían generar problemas.

- JavaScript ofrece acceso a Node.js para muchas bibliotecas útiles y, por supuesto, Node.js brinda la posibilidad de compartir código entre navegador y servidor ya que ambos usan código JavaScript. Basado en esto, los desarrolladores han creado MEAN, una web de JavaScript que cuenta con un conjunto de aplicaciones que consta de MongoDB (una base de datos controlada por JavaScript), Express y Angular (un marco de JavaScript front-end)

1.3.10 REACT NATIVE

React Native es un framework de JavaScript para escribir aplicaciones nativas y móviles en tiempo real, se basa en React, la biblioteca de JavaScript de Facebook para la construcción de interfaces de usuario, pero en lugar de apuntar al navegador, apunta a prototipos móviles. En otras palabras, los desarrolladores web ahora pueden escribir aplicaciones móviles que se ven y se sienten verdaderamente locales, todo desde la comodidad de una biblioteca de JavaScript que ya se conoce.

React Native en vez de incrementar una web híbrida o en HTML5 genera una aplicación nativa con un rendimiento muy similar al nativo, permitiendo, que el compilador que maneja React Native va a convertir los documentos en elementos nativos de la interfaz para Android y para iOS, lo que alcanza que dichas aplicaciones tengan una apariencia muy semejante a las aplicaciones nativas, pues su rendimiento es casi similar debido a que fundamentalmente se está generando una interfaz nativa [17].

Dentro del funcionamiento de React Native y las aplicaciones desarrolladas se evidencia que está formado por 3 capas que son: [17]

- Capa nativa: maneja los elementos nativos de la Interfaz de Usuario y procesa los eventos y consumos que el usuario solicita.
- Capa JavaScript: genera un código JavaScript, el cual se encarga de la lógica de la aplicación, funcionalidades y la estructura de la interfaz de usuario.

- Capa bridge: Esta capa se asemeja a la de un controlador de comunicaciones asincrónicas, envía mensajes entre capas de manera optimizada.

Muchas de las aplicaciones más comunes y usadas han sido desarrolladas bajo React Native tales como [18]:

- Skype.
- Facebook.
- Uber Eats.
- Coinbase.
- Wix.
- SoundCloud.
- Airbnb.
- Instagram.
- Bloomberg.
- Walmart.

1.3.11 JAVASCRIPT

Java Script es un lenguaje de programación que se caracteriza especialmente por trabajar de la mano con React Native, ya que este se basa en una biblioteca que contiene múltiples interfaces y métodos que permiten el desarrollo de aplicaciones y ahora con React Native en vez de apuntar a un navegador, este apunta a prototipos móviles, logrando que los desarrolladores web puedan realizar aplicaciones móviles que se sienten nativas [18].

Las aplicaciones pertenecientes a React Native están escritas en un lenguaje que combina a JavaScript y marca al estilo XML. JavaScript es una gran ventaja sobre React Native, puesto que React Native presenta interfaces JavaScript para API de prototipo teniendo como resultado el acceso a la cámara del teléfono, a su ubicación, etc, destacándose de otras aplicaciones multiprototipo [18].

1.3.12 SAAS (Syntactically Awesome Style Sheets - hojas de estilo en cascada)

Un procesador CSS es una herramienta que permite organizar, generar a escribir, estructurar y organizar el código CSS, es de gran ayuda para diseñar aplicaciones CSS de gran tamaño llegando a ser una de las herramientas más usadas en la actualidad.

Cabe recalcar que SAAS es una herramienta de desarrollo y no es considerado como una extensión de CSS, lo que se refiere es que cuando se usa un paquete

web o de módulos para la construcción de un código se compila los archivos SAAS, se conoce que varias bibliotecas y marcos CSS modernos como Bootstrap están escritos en código SAAS. Además, funciona con marcos fronted como React, por esto es que a la vez se lo puede usar en proyectos React Native.

SAAS en la web funciona de la siguiente manera: debido a que React Native funciona con objetos de estilo mas no con CSS, sus archivos SAAS también se convierten en objetos de estilo no visible. Esto le da una ventaja significativa de escribir CSS sin procesar para diseñar sus componentes React Native. Su proyecto React Native compila archivos SAAS en tiempo de ejecución y los convierte en objetos JavaScript normales.

El proyecto React Native o paquete react-native-SAAS transformer permite configurar y usar SAAS en cualquier tipo de proyecto React Native. Node-SAAS es otro paquete que se utiliza como una dependencia para conceder el enlace de Nodo pasándolo a SAAS [19].

1.3.13 METODOLOGÍA KANBAN

Los proyectos de todos los tamaños y tipos requieren una gestión estructurada para completarse eficientemente, correctamente y a tiempo. Si bien algunos procesos son simples, otros son muy complejos e involucran un conjunto de subprocesos. Si se omite algún paso, todo el proyecto puede verse afectado. La gestión del flujo de trabajo y el flujo de información es importante.

Kanban es uno de los marcos de trabajo más efectivos para administrar los procesos en cualquier proyecto. Este concepto está diseñado para brindar a las personas un flujo de trabajo más eficiente, al optimizar todas las tareas involucradas. Con un proceso más eficiente, asegura que los proyectos sean completados de forma rápida y eficaz [20].

Kanban es una palabra fusión con raíces japonesas. La palabra "kan" significa visual, y el significado de la palabra "ban" es tarjeta. Como término japonés común, la combinación de las palabras se usa comúnmente para referirse a vallas publicitarias o tableros visuales. Con cualquier software programa o sistema de gestión utilizando un concepto Kanban, el uso de tableros o, las tarjetas son una

técnica de visualización que ayuda a identificar el progreso, las tareas y detalles de la organización en una ubicación central [21].

Kanban tiene la capacidad de reducir el tiempo empleado en cada tarea y aumentando así los índices de calidad y eficacia notablemente al limitar la cantidad de trabajo de forma estricta, esta se puede implementar en 4 pasos [22]:

- a) Definir el flujo de trabajo dentro del proyecto.
- b) Visualizar las fases del ciclo de trabajo (producción).
- c) Acabar una tarea antes de iniciar la siguiente.
- d) Controlar el flujo de actividades.

2. METODOLOGÍA

En el capítulo anterior se indican los objetivos, alcance, marco teórico del proyecto. En este capítulo se detallan los principios fundamentales para el correcto desarrollo del proyecto. Dentro del proyecto se diseña y se implementa una aplicación web a la que se podrá acceder desde un computador o un equipo móvil.

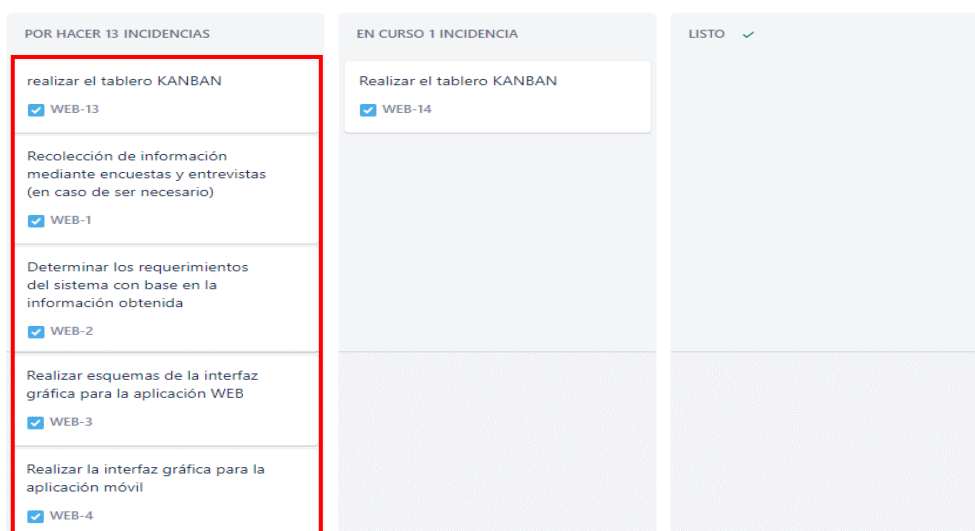
2.1 DESARROLLO

Dentro de este apartado se podrá encontrar el diseño de la base de datos, los esquemáticos que componen la interfaz gráfica de la aplicación web, los modelamientos necesarios para los diferentes usos, y la lógica de los procesos.

Como se mencionó anteriormente la metodología utilizada para el desarrollo es KANBAN, por lo tanto, se implementa el tablero con las tareas a realizar para el respectivo diseño del sistema, de la misma forma podrá encontrar los resultados de las pruebas de funcionalidad.

2.1.1 ELABORACIÓN DE TABLERO KANBAN

En la figura 2.1 se observa las tareas específicas para el diseño del sistema, implementación y pruebas del mismo, clasificadas por los colores rojo, amarillo y verde respectivamente.



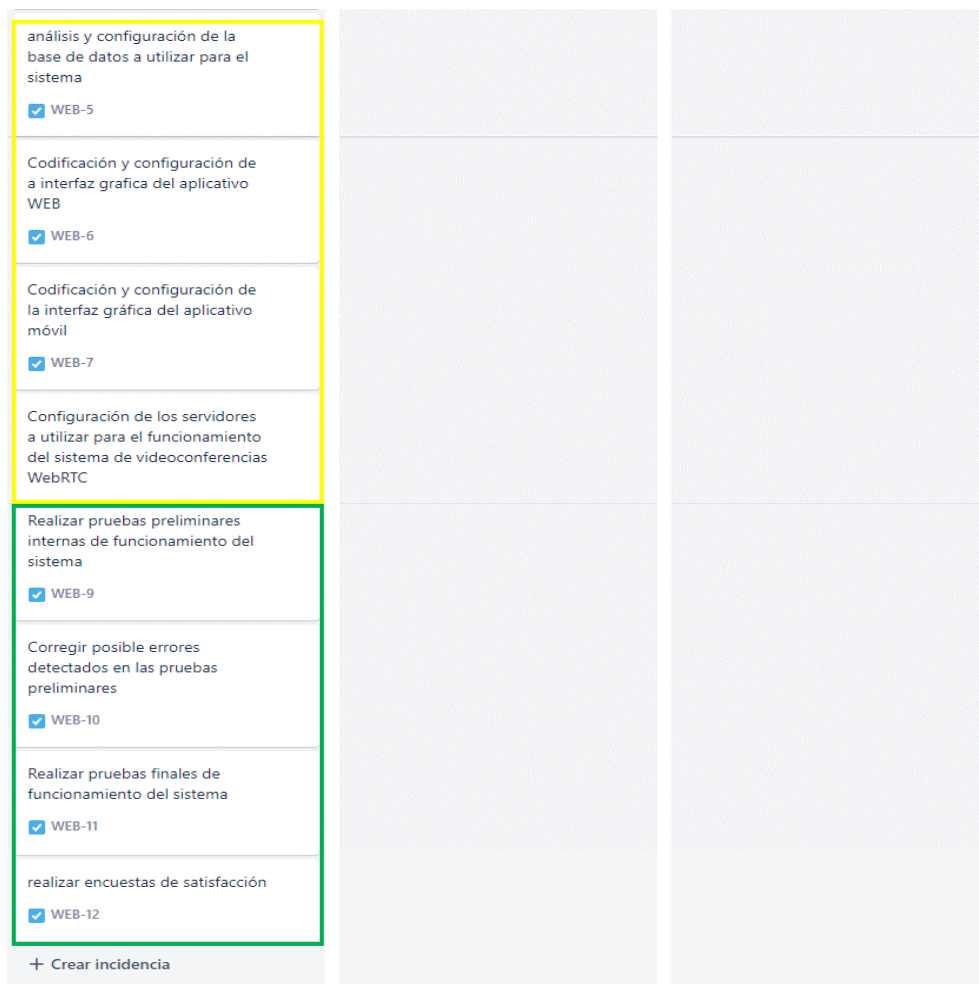


Figura 2.8. Tablero KANBAN de tareas a realizar para el desarrollo del sistema.

2.1.2 REQUERIMIENTOS

Los requerimientos del sistema serán determinados mediante la recolección de datos en las encuestas del ANEXO A que se realizó a los docentes y autoridades del Instituto Superior Tecnológico Sucre (ISTS).

Sin embargo, existen requerimientos que el prototipo debe cumplir, mismos que se describen a continuación:

NO FUNCIONALES

Los requerimientos no funcionales son obtenidos a través de las características necesarias para que el sistema funcione, en este caso se han definido los siguientes:

- Los navegadores compatibles deben ser actualizados periódicamente en conjunto con el sistema operativo.
- El aplicativo Web no necesita realizar cambios en la configuración de los parámetros en los equipos (desactivación de firewall, antivirus entre otros).
- Los errores registrados durante la operación del sistema deben enviar mensajes al administrador general con las notificaciones.

FUNCIONALES

- El prototipo debe contar con una interfaz sencilla e intuitiva tanto para la aplicación web como para la móvil.
- La seguridad de las comunicaciones a través de este sistema utilizará los protocolos HTTPS.
- El usuario deberá ser capaz de crear una cuenta para acceder al prototipo y utilizar sus herramientas, misma que será registrada en la base de datos del sistema.
- El prototipo de videoconferencia deberá contar con funciones que permitan el control y la administración de la transferencia de datos (video llamadas, intercambio de información, mensajería instantánea, y demás que sean determinados por el ISTS).
- La autenticación de los usuarios debe darse por medio del correo institucional registrado en la base de datos del ISTS.
- El prototipo debe permitir que el usuario se registre mediante un rol específico sea este de moderador conferencista o usuario participante para la asignación de atributos y privilegios.
- El prototipo permitirá que el usuario pueda modificar los datos de presentación (nombre de usuario).
- El prototipo permitirá la comunicación entre los usuarios de la sesión mediante mensajería instantánea (chat).
- El prototipo permitirá la compartición de pantalla de manera única y exclusiva para quien la solicite, siempre y cuando no exista una compartición previa.
- El prototipo permitirá recibir en tiempo real notificaciones de los eventos durante la sesión.

2.1.3 ROLES Y PERMISOS

Una vez determinados los diferentes requerimientos, se procede a la determinación de roles, atributos y privilegios para los dos tipos de usuarios.

TIPOS DE USUARIOS

Se encuentran tres tipos de usuarios para el sistema en desarrollo.

- Moderador conferencista.
- Usuario participante.
- Administrador general.

ASIGNACIÓN DE ROLES Y PERMISOS

Dentro del sistema se realiza la definición de los roles para la correcta interacción del usuario con el sistema.

Moderador conferencista: encargado de la gestión del sistema de videoconferencia, capaz de permitir el ingreso a los usuarios participantes, establecer restricciones de funcionalidad, y privilegios de iniciar, terminar, grabar y transmitir en vivo la sesión de videoconferencia.

Usuario participante: participantes de las sesiones de videoconferencias a las que deben ser admitidos mediante contraseña, supeditados a las configuraciones y restricciones del moderador conferencista, por lo tanto, sus atributos y privilegios son limitados.

Administrador general: dentro del sistema de videoconferencia, existe un único usuario que tendrá el rol **de administrador general**, mismo que se encarga de ejecutar y supervisar tareas como; interacción del sistema de las bases de datos entre el prototipo y el ISTS, monitoreo del funcionamiento de los servidores, de la misma manera este tendrá privilegios de realizar la creación, modificación, actualización y eliminación de usuarios, CRUD, previo a la autorización del ISTS.

2.1.4 CASOS DE USO

Ya con los roles establecidos en conjunto con sus atributos y privilegios se procede a realizar un diagrama de casos de uso para cada uno de los actores, mismos que se determinan mediante las funcionalidades requeridas.

En este apartado se detallan las funciones u opciones que el usuario puede realizar dentro del sistema, dependiendo del rol asignado.

Cabe mencionar que ambos tipos de usuario tienen procesos y usos en común como: el inicio de sesión, visualización de participantes, comunicación entre usuarios ya sea grupal o individual, recepción de notificación, etc.

A continuación, la figura 2.2 y 2.3, presentan un diagrama de los casos de uso del moderador conferencista, usuario participante y administrador general.

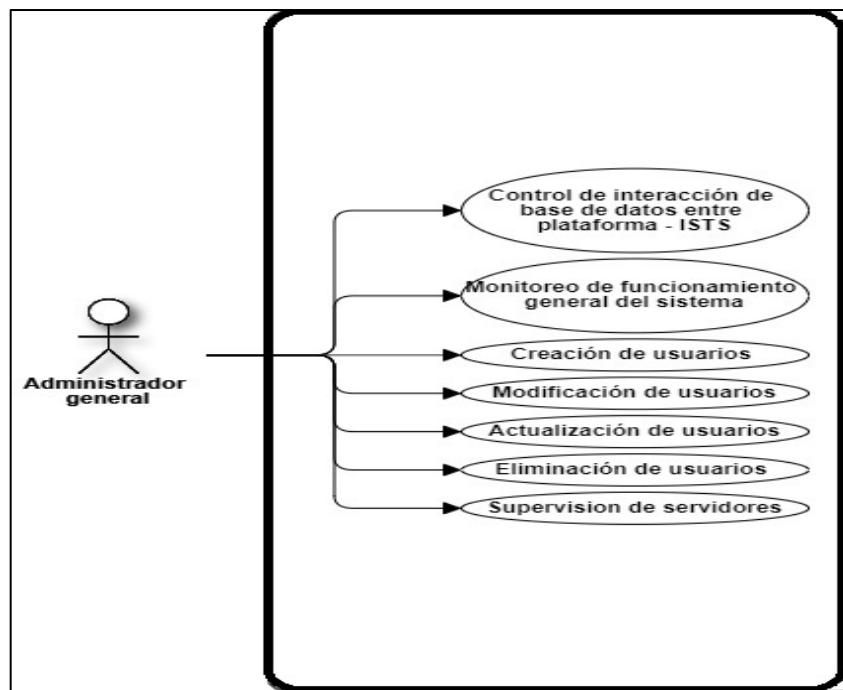


Figura 2.9. Diagrama de casos de uso de Administrador General.

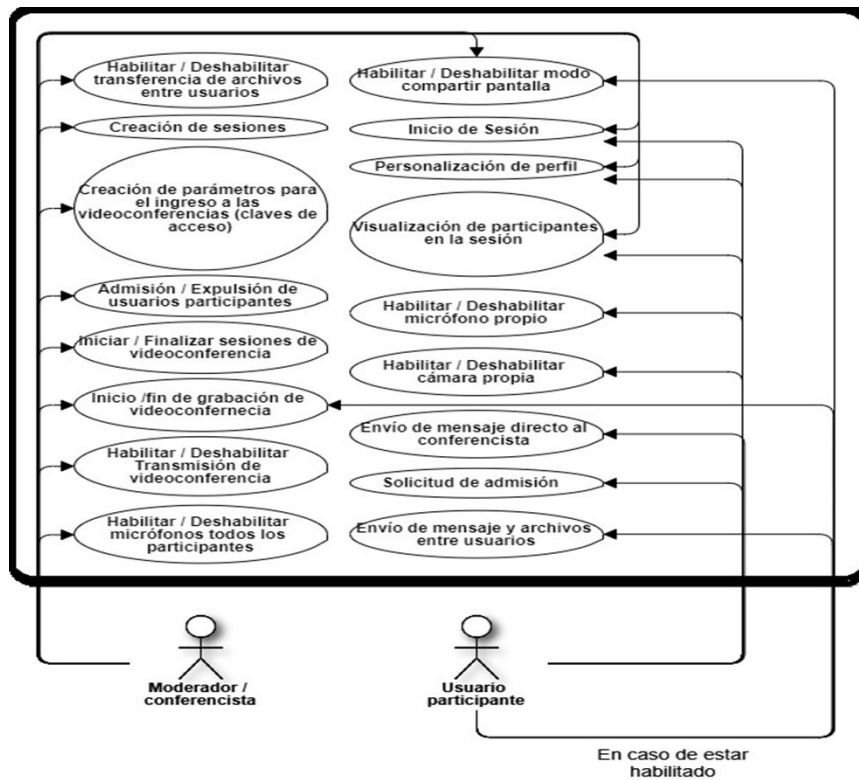


Figura 2.10. Diagrama de casos de uso de Usuario/Moderador.

2.1.5 INTERFAZ GRÁFICA

El sistema deberá contener una interfaz gráfica que permita la interacción con el usuario, para esto es necesario la implementación de módulos cuya funcionalidad dependerán del rol asignado por el sistema al momento de iniciar sesión [23].

A continuación, se muestran los bosquejos más importantes del prototipo desarrollado:

En la figura 2.4. se representa la pantalla de inicio con un fondo de pantalla representativa para el Instituto Superior Tecnológico Sucre, en el cuadro de texto se debe ingresar el nombre de sala de videoconferencia a la cual requiere unirse.

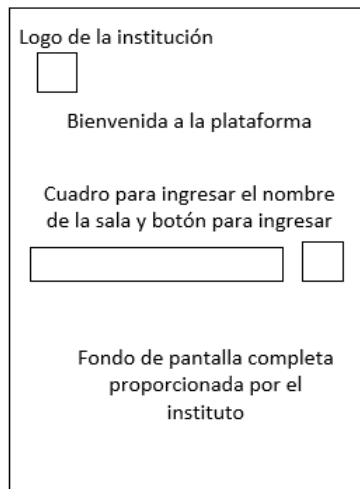


Figura 2.11. Sketch de pantalla de bienvenida.

En la figura 2.5 se muestra una pantalla de confirmación para acceder al menú de inicio de sesión o cancelar para regresar a la página de bienvenida.

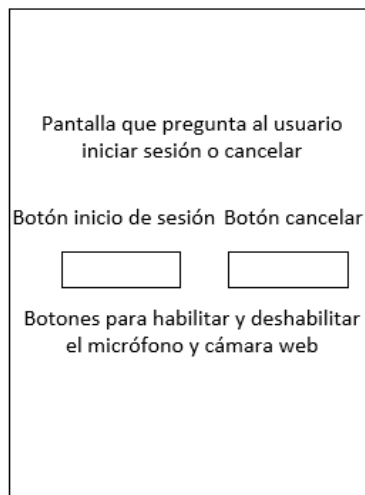


Figura 2.12. Sketch de confirmación para inicio de sesión.

En la figura 2.6 se muestra la pantalla para el ingreso de nombre de usuario y contraseña. El Administrador es el único que puede modificar las contraseñas de los usuarios registrados en el sistema. Al ser un prototipo, el sistema no proporciona la recuperación de contraseñas.

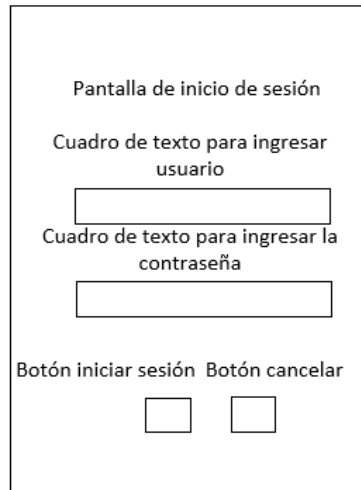


Figura 2.13: Sketch de pantalla de ingreso de credenciales del usuario.

En la figura 2.7 se muestra los parámetros y características de la sala de videoconferencia.

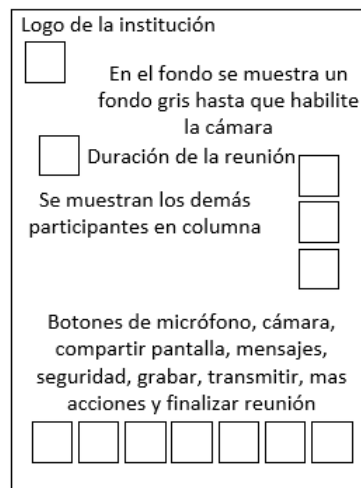


Figura 2.14: Sketch de pantalla de opciones en la sesión.

A continuación, se describen a detalle los módulos que conforman el sistema WebRTC para videoconferencias desarrollado en el proyecto.

- **Registro:** este módulo es exclusivo del administrador y adicional a los indicados en el alcance del proyecto. El módulo permitirá que el

administrador pueda registrar nuevos usuarios en la base de datos del prototipo, ingresando los siguientes parámetros: nombre de usuario y contraseña, que serán las credenciales para el ingreso al sistema.

- **Autenticación:** módulo necesario para la validación de credenciales creadas durante el registro y tener acceso al sistema de videoconferencias.
- **Chat:** permite la comunicación y compartición de mensajes entre todos los usuarios que se encuentren en la sesión iniciada por el moderador.
- **Perfil:** Permite la edición del nombre de usuario.
- **Notificación:** Este módulo permite a todos los usuarios la visualización de notificaciones de manera automática durante todo el tiempo que se encuentre dentro del sistema, por ejemplo: errores de autenticación, restricciones hechas por el moderador conferencista, etc.
- **Trabajo:** dentro de este módulo se establece las funcionalidades exclusivas para moderadores, como lo es el permitir la admisión de un usuario mediante contraseña, la expulsión del mismo y el privilegio de habilitar / deshabilitar las diferentes funciones, que se muestran el cuadro de casos de uso para usuarios participantes.

2.1.6 LÓGICA DE PROCESOS

En este apartado se explicará el funcionamiento del sistema mediante el uso de diagramas de actividades y de secuencia del sistema.

Proceso de registro y autenticación de usuarios.

A continuación, en la figura 2.8 se presenta el diagrama de actividades para el proceso de registro y autenticación de usuarios al prototipo.

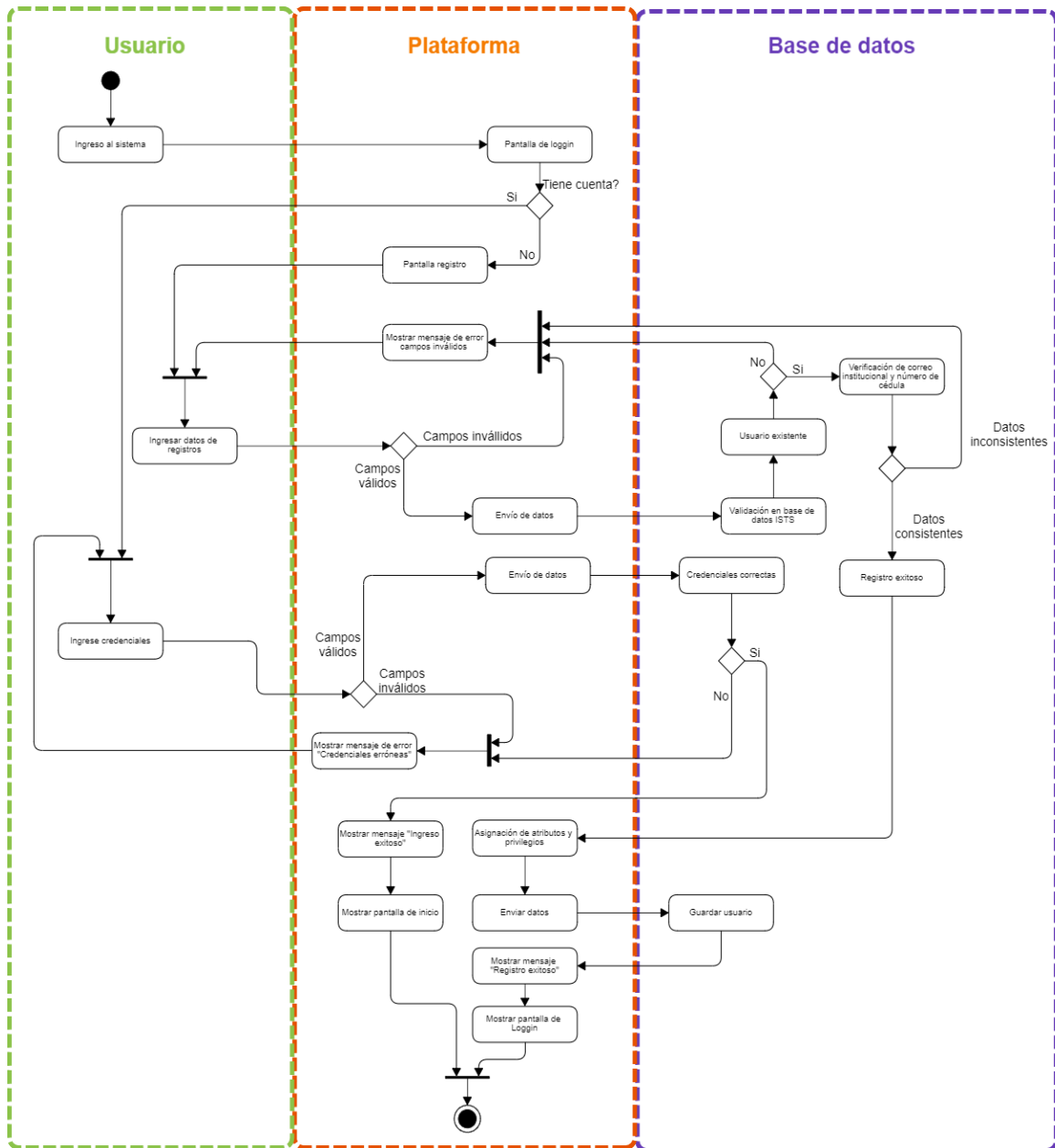


Figura 2.15. Diagrama de actividades del proceso de registro y autenticación de usuarios.

El diagrama anterior ilustra el proceso de inicio de sesión y registro de un nuevo usuario. Una vez que el usuario ingrese al sistema se presentará la pantalla de login (acceso a la prototipo) aquí el usuario deberá colocar las credenciales registradas, si los campos no se llenan de manera correcta el sistema presentará un mensaje de error volviendo a presentar la página, si los campos llenados de usuario y contraseña se completan correctamente se verificará en la base de datos que estos estén registrados, de ser el caso se presentará un mensaje de ingreso exitoso y a

continuación se mostrará la pantalla de inicio del aplicativo, dado el caso de que el usuario no cuente con credenciales deberá realizar el registro. Las figuras 2.9 y 2.10 muestran los diagramas de secuencia del bloque descrito anteriormente referente al registro de usuarios e inicio de sesión respectivamente.

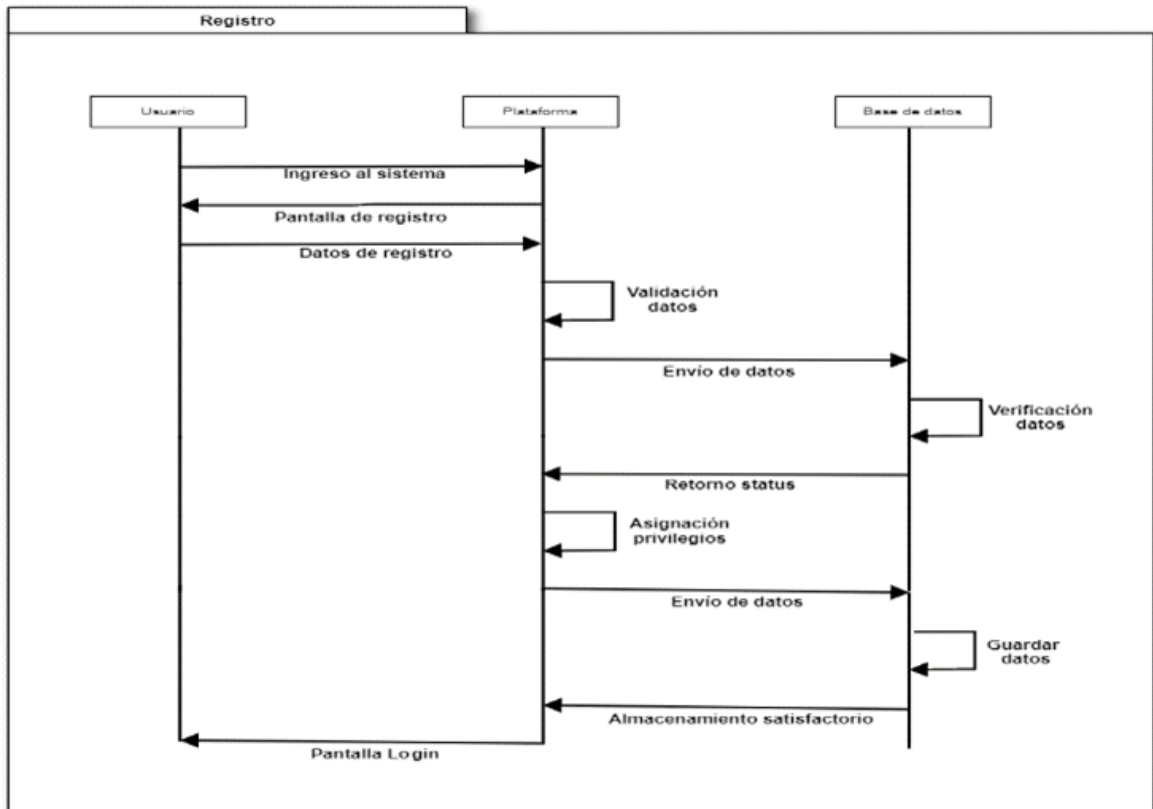


Figura 2.16. Diagrama de secuencia del proceso registro de usuario.

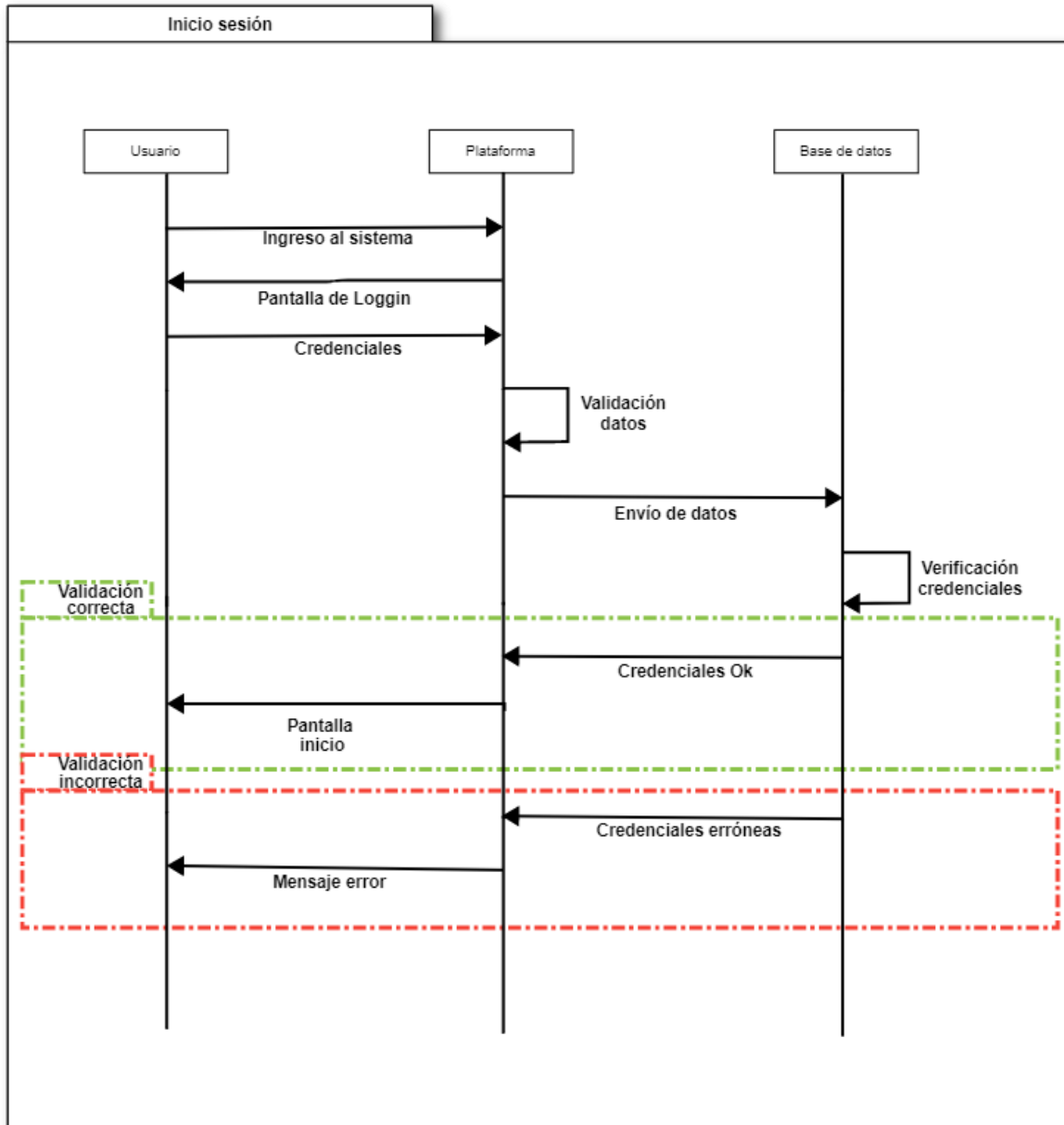


Figura 2.17. Diagrama de secuencia del proceso inicio de sesión

En estos diagramas de secuencia se explica la interacción de los datos entre el usuario, la plataforma diseñada y la base de datos, en la figura 2.9 para el registro de datos, inicialmente la interacción es entre el usuario y la plataforma, es decir el usuario debe ingresar sus credenciales en la plataforma, una vez que el usuario complete los campos y de click en “ingresar” la interacción pasa a ser entre la plataforma y la base de datos. En este caso el proceso es interno, el usuario no deberá realizar ninguna acción y esperar a que la plataforma le devuelva el “status”, la validación de datos y asignación de atributos se genera de forma automática según los datos ingresados.

En la figura 2.10 se presenta la secuencia de la validación de datos cuando un usuario ya se encuentra registrado. En este caso existen dos posibles respuestas, “usuario correcto” y “usuario incorrecto”. El proceso inicia con el usuario accediendo al prototipo. Luego de ingresadas las credenciales comienza el proceso interno para realizar una búsqueda en la base de datos, si los datos son encontrados; se dará el acceso con los atributos asignados al usuario, caso contrario el prototipo presentará una mensaje de “usuario incorrecto”, se vaciarán los campos, y se permitirá el ingreso nuevamente para realizar una nueva búsqueda.

Proceso del módulo de chat

La figura 2.11 muestra el diagrama de actividades del módulo “chat”, básicamente el usuario una vez que haya accedido al sistema ingresando sus credenciales tendrá la opción de comunicarse con otros usuarios (para los usuarios participantes debe estar habilitado el servicio por parte del moderador, caso contrario, solo podrá comunicarse mediante chat con el moderador) mediante un servicio de mensajería instantánea presionando el indicador “chat” en la pantalla de inicio.

Dentro de la ventana chat el usuario deberá seleccionar con quien desea intercambiar mensajes, en ese momento el sistema presentará la conversación, se escribe el mensaje, mismo que el prototipo enviará al servidor de base de datos para almacenarlo, procesarlo y finalmente actualizar la conversación.

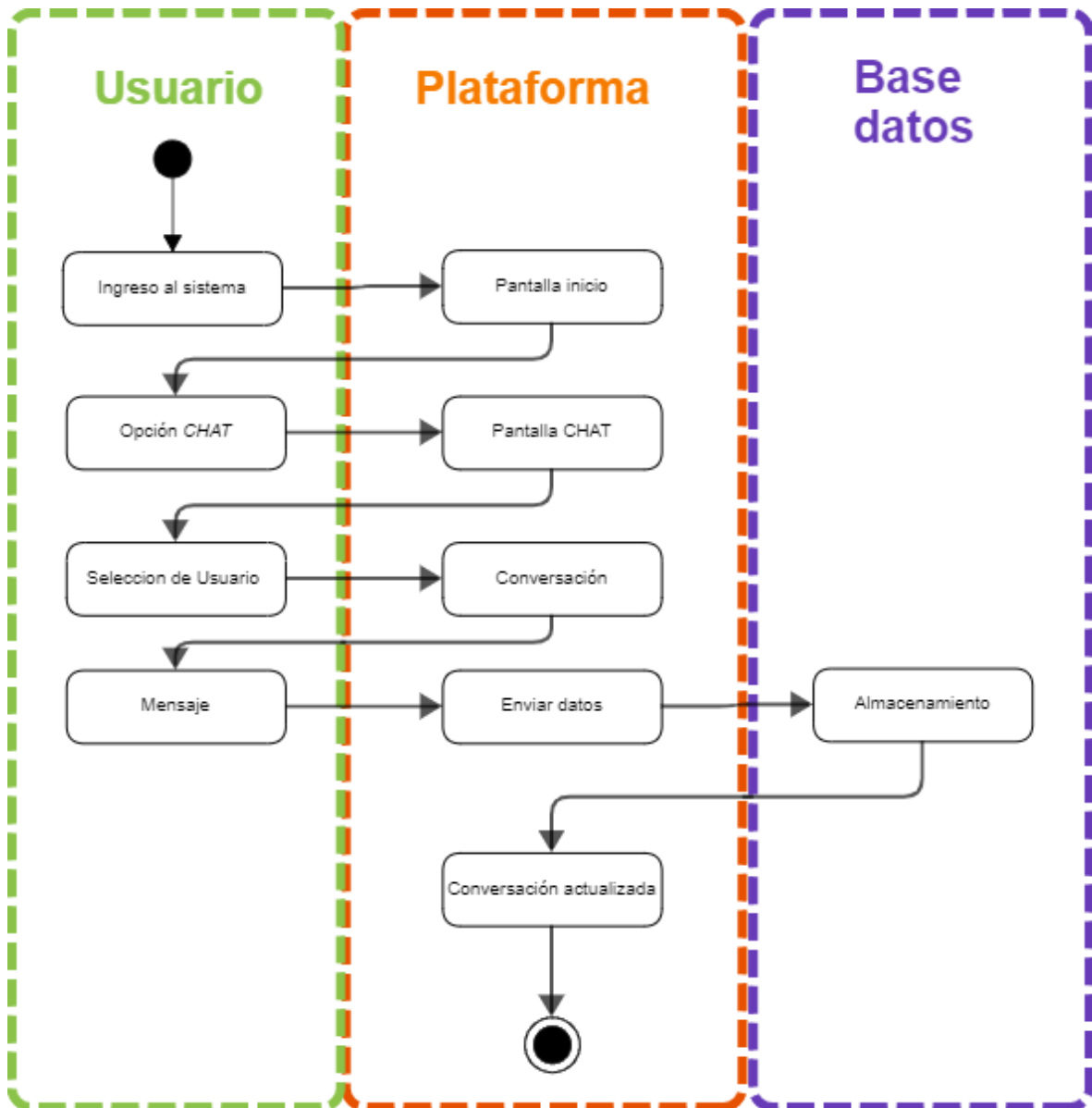


Figura 2.18. Diagrama de actividades del proceso chat.

La figura 2.12 muestra el diagrama de secuencia del proceso chat entre usuarios de sesión. La interacción de este proceso inicia con la selección del usuario con quien se va a utilizar el servicio de mensajería, llena el campo "mensaje" y presiona el botón "enviar", una vez presionado este botón se iniciará el proceso de transferencia de información entre los usuarios, pasando por la base de datos para almacenar el mensaje, y finalmente actualizando los campos en ambos usuarios, el proceso se repite en sentido contrario para recibir una respuesta del segundo usuario al mensaje enviado por el primer usuario.

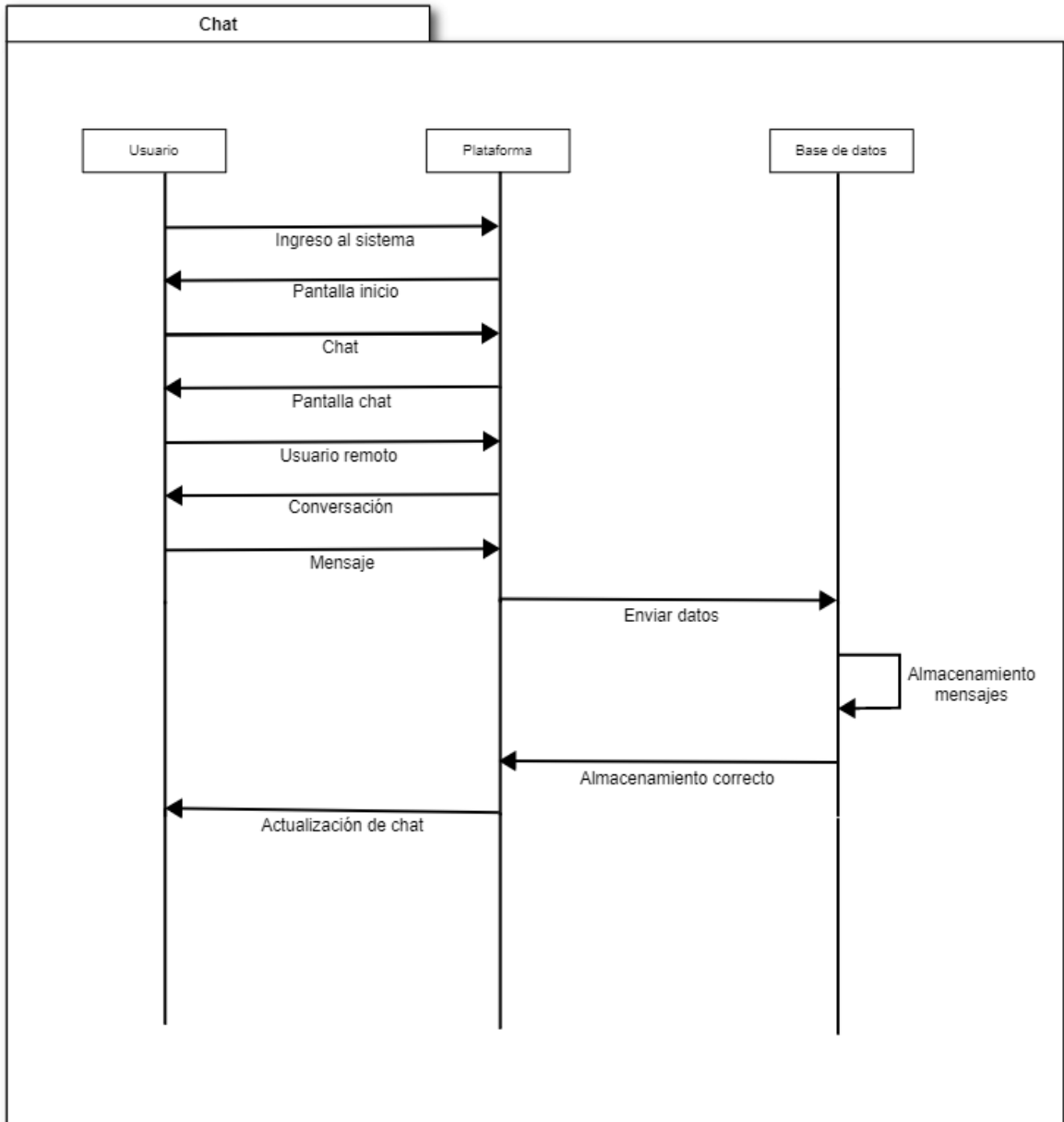


Figura 2.19. Diagrama de secuencia del proceso chat.

Proceso Perfil

Dentro de este apartado se describe el funcionamiento de este proceso que permite al usuario cambiar o actualizar el nombre de usuario que se muestra a los demás participantes, los campos de nombre de usuario y contraseña serán los únicos que no pueden ser modificados, debido a que con estos se realiza la validación inicial para poder registrarse en el sistema.

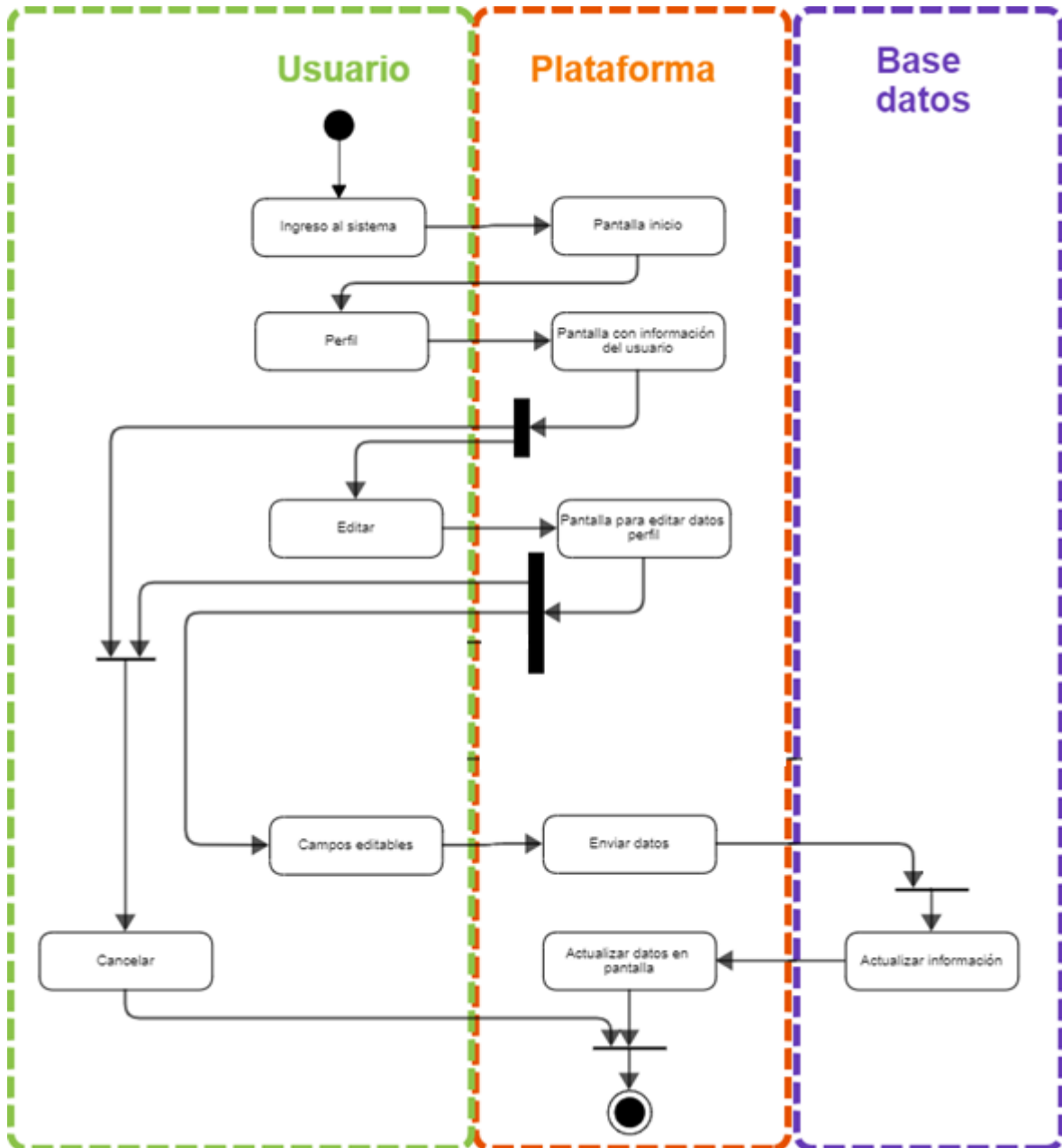


Figura 2.13. Diagrama de actividades del proceso para el módulo perfil.

La figura 2.13 explica el funcionamiento del bloque donde el usuario al momento de seleccionar “perfil” tiene la opción de editar el nombre de usuario que se muestra a los demás participantes, en cualquier momento del proceso el usuario puede realizar la cancelación del proceso sin realizar modificación alguna en los registros previamente guardados al momento de realizar el proceso inicial de registro de usuario, y en la figura 2.14 se muestra el diagrama de secuencia. Este proceso se inicia con el inicio de sesión del usuario, ingresando las credenciales registradas y

validadas por el sistema, una vez dentro del sistema selecciona la opción “modificar”, lo que permitirá al usuario realizar cambios en los datos.

Este proceso le permite al usuario reingresar datos en los campos ya registrados anteriormente, dentro de este la plataforma presentará información extraída desde la base de datos con la opción de realizar cambios en ellos, el usuario vuelve a llenar los campos y al momento de “guardar” los datos, esto serán enviados hacia la base de datos reemplazando a los datos anteriormente presentados, guardando nuevamente la información y asignando atributos, según los datos ingresados.

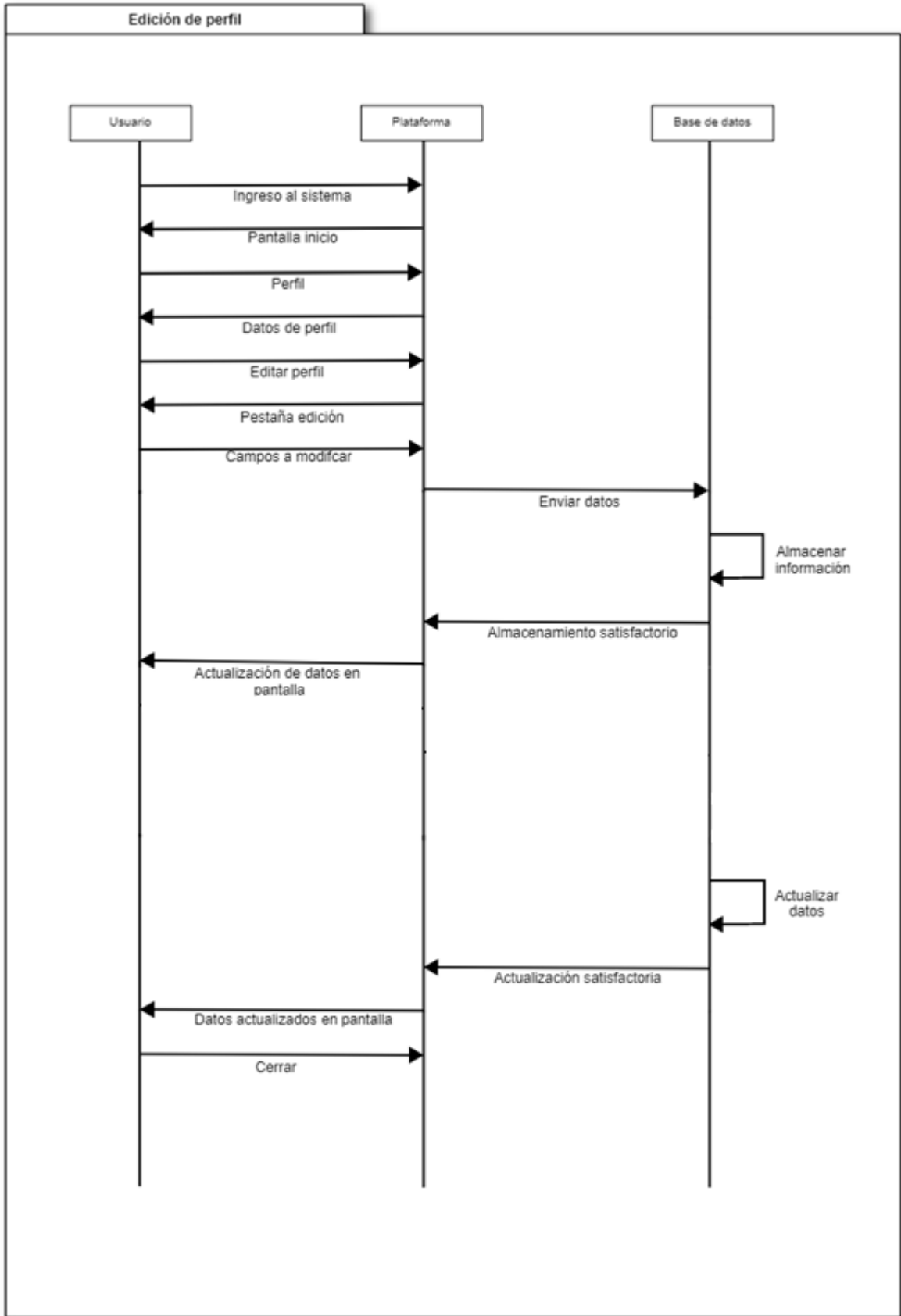


Figura 2.14. Diagrama de secuencia del proceso edición de perfil

Proceso notificaciones

En las figuras 2.15 y 2.16 se muestran los diagramas de actividades y de secuencia respectivamente. En estos se explican cómo funciona este módulo, el sistema tiene la capacidad de enviar notificaciones en tiempo real de los eventos que acontecen durante la sesión de videoconferencia, para poder visualizarlas el usuario debe dirigirse al icono “notificación” el cual mostrará una pestaña con todas las eventualidades sucedidas.



Figura 2.15. Diagrama de actividades del proceso notificación.

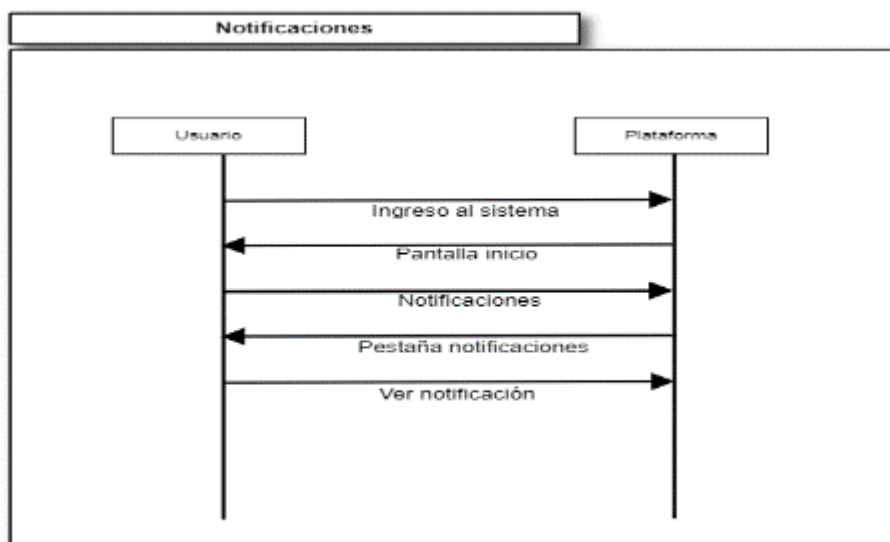


Figura 2.16. Diagrama de secuencia del proceso notificación.

Proceso Trabajo

En este apartado se detalla cómo trabaja el módulo en función de los requerimientos del conferencista moderador, figuras 2.17 y 2.18. Recordando que este tiene los privilegios de restringir ciertas funcionalidades dentro del prototipo para los usuarios participantes, como la creación de clave para el acceso a la videoconferencia, habilitar / deshabilitar micrófonos, modo de compartición de pantalla, grabaciones de videoconferencia, y demás mencionadas en el ítem roles y permisos.

Creación de sesión

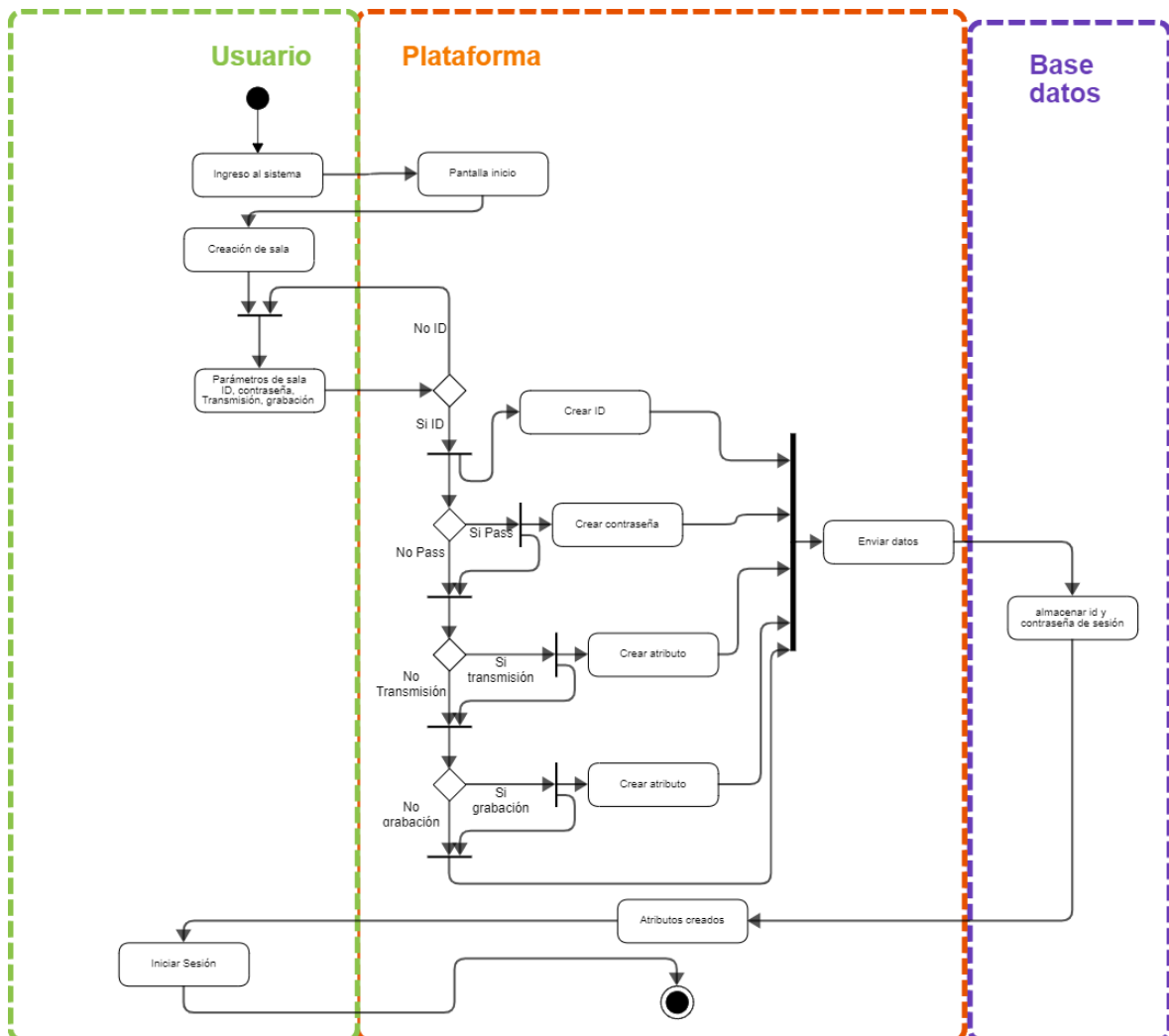


Figura 2.17. Diagrama de actividades para la creación de sesión.

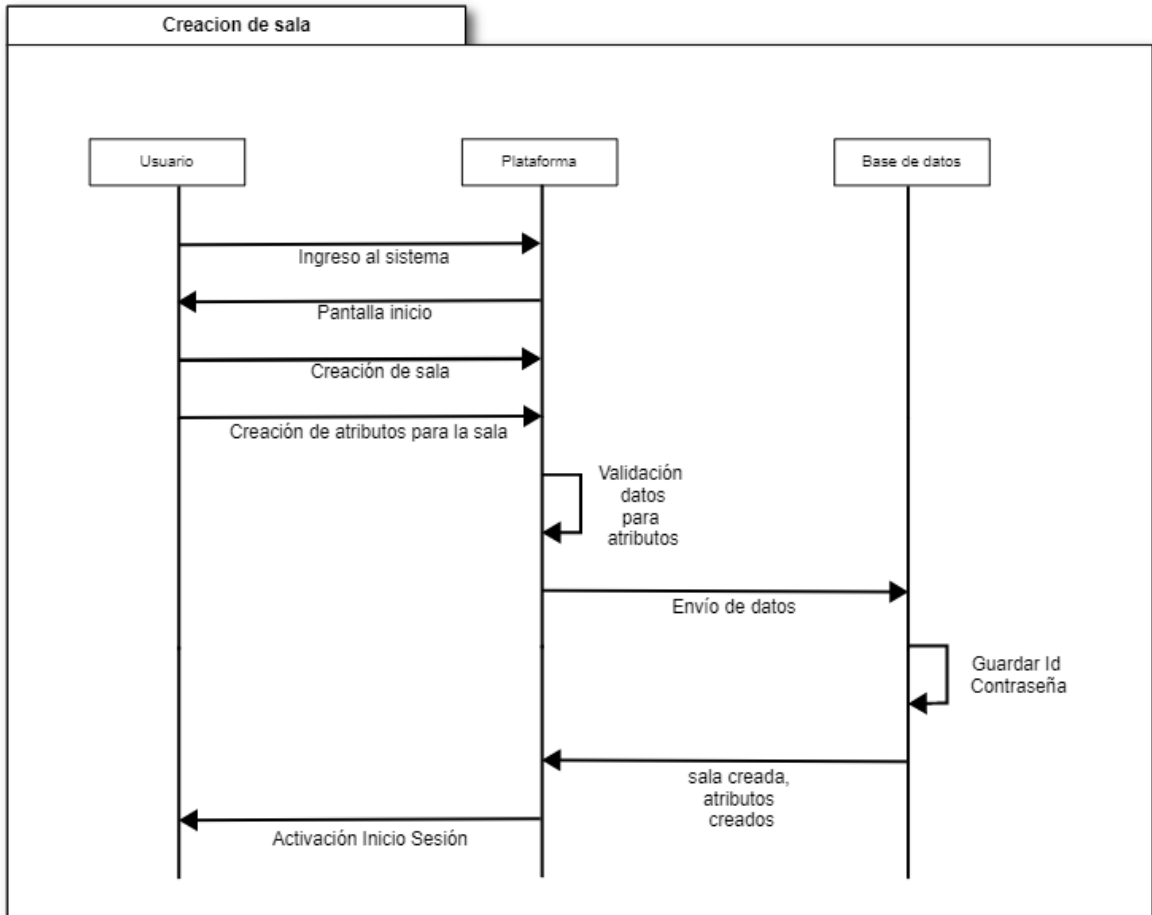


Figura 2.18. Diagrama de secuencia para la creación de sesión.

Diagrama de Clases

El Diagrama de clases permite representar la estructura de un sistema al modelar sus clases, atributos, operaciones y relaciones entre objetos.

En la figura 2.19. se representa el Diagrama de Clases de acuerdo al modelado del prototipo, se muestra las clases con sus respectivos métodos y atributos.

El Diagrama de Clases está diseñado con las siguientes clases: la clase UsuarioSTS representa la información de los usuarios que se van a enviar al servidor de videoconferencia en esta clase se puede crear, actualizar, consultar, eliminar y transferir usuarios, la clase Prosody se encuentra en el servidor de autenticación y se encarga de autenticar usuarios y los servidores que conforman la infraestructura del servidor de videoconferencia, la clase Jicofo se encarga de unir a los usuarios a las distintas salas de videoconferencia también se encarga del

balanceo de carga en cuanto al uso de recursos, la clase Jibri grabar y transmitir la sala de videoconferencia, la clase Videobridge se encargan transmitir audio y video de los participantes en las salas de videoconferencia, la clase Sala se encarga de desplegar las opciones cámara, micrófono, compartir pantalla, grabar sala, transmitir la sala, ayuda, notificaciones, chat y salir de la videoconferencia, la clase TipoUsuario se encarga de habilitar y deshabilitar permisos según el tipo de rol del usuario moderador o invitado, la clase Chat permite enviar y recibir mensajes en el chat general como en los chat privados de los usuarios.



Figura 2.19. Diagrama de clases.

2.2 IMPLEMENTACIÓN DEL SISTEMA

El desarrollo de la implementación del prototipo consiste en convertir todas las características mencionadas anteriormente en un aplicativo web. A continuación, se presenta el desarrollo del sistema desde su diseño hasta la realización de las pruebas.

2.2.1 APLICACIÓN WEB

Para la creación de la interfaz gráfica y su codificación se utilizó el software “*Visual Studio Code*”; el software se instaló en una máquina virtual con distribución Linux y la virtualización se hizo utilizando un computador de escritorio. En la tabla 2.1 se muestran las características de las máquinas utilizadas.

Especificaciones	Máquina Real	Máquina virtual
Sistema Operativo	Windows 11 pro	Ubuntu 20.04
Memoria RAM	16 GB	8 GB
Procesador	Intel i7 8700	Intel i7 8700
Número de núcleos	12	12
Almacenamiento	1 TB	50 GB

Tabla 2.1. Especificaciones técnicas de máquinas usadas para la virtualización.

En la figura 2.20 se presenta la máquina virtual de desarrollo Ubuntu instalada en VMware Workstation donde se muestra el programa Visual Studio Code utilizado para la codificación y pruebas del servidor web mismo que será replicado en el servidor del instituto.

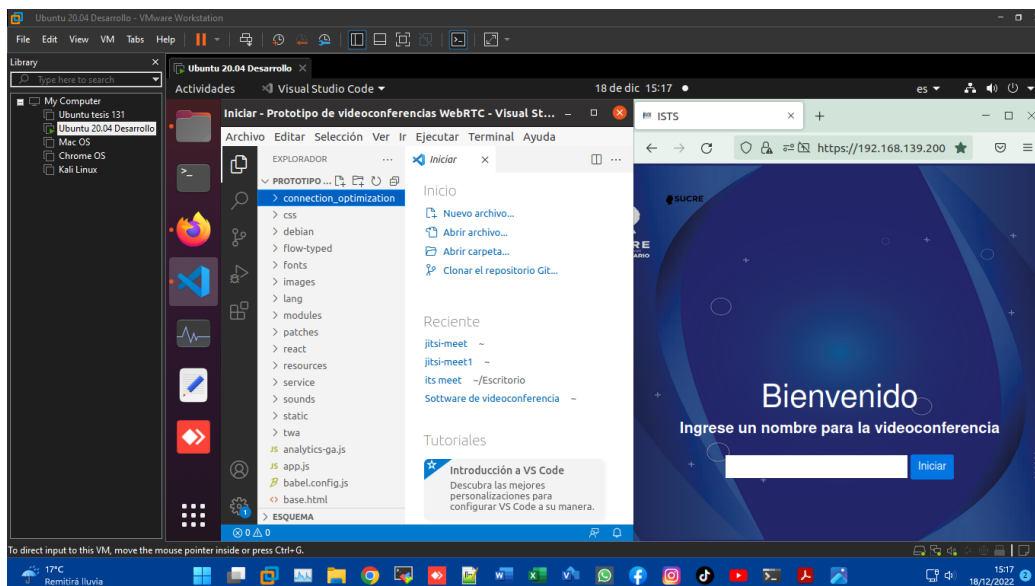


Figura 2.200. Pantalla de bienvenida tomada en las pruebas preliminares del sistema.

2.2.2 BASE DE DATOS

Es necesaria la implementación de una base de datos denominada `db_usuarios_reuniones`, misma que tiene la información de inicio de sesión de los usuarios y son almacenados en un prototipo creado en MySQL, herramienta virtualizada que se instaló en uno de los servidores físicos del ISTS, de la misma manera como se hizo con la virtualización para el desarrollo de la aplicación web, con el uso del software VMWare para la virtualización de los equipos, figura 2.24.

Las características de los equipos utilizados se presentan en la tabla 2.2.

Especificaciones	Servidor real Dell R440	Servidor virtual
Sistema Operativo	VMWARE ESXI 6.7.0	Ubuntu 20.04
Memoria RAM	79.5 GB	16 GB
Procesador	Xeon Bronze 3106	Xeon Bronze 3106
Número de núcleos	12	8
Almacenamiento	1 TB	50 GB

Tabla 2.2: Especificaciones técnicas de equipos usados e implementados para virtualización de servidor de almacenamiento dentro del ISTS.

Para acceder a la configuración del servidor de videoconferencia, debido a que no se tiene acceso físico a este, se lo hace de forma remota mediante un servicio SSH a través del uso de la dirección IP pública.

En la figura 2.21. se muestra la pantalla de administración de los servidores virtuales VMware ESXi donde se creó el servidor virtual Ubuntu.

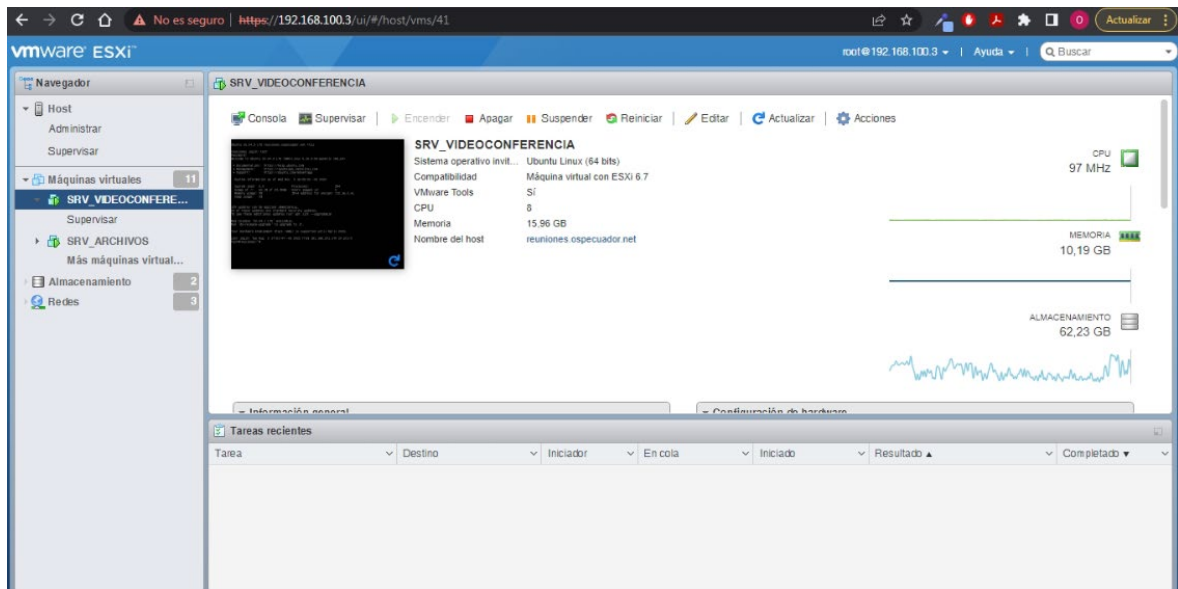


Figura 2.21. Pantalla de máquina virtual Ubuntu 20.0 instalada en VMWare ESXi para el servidor de videoconferencia.

En la figura 2.22 se muestra el ingreso a la máquina virtual mediante el protocolo SSH donde se muestra las características del servidor como RAM, disco duro y procesador.

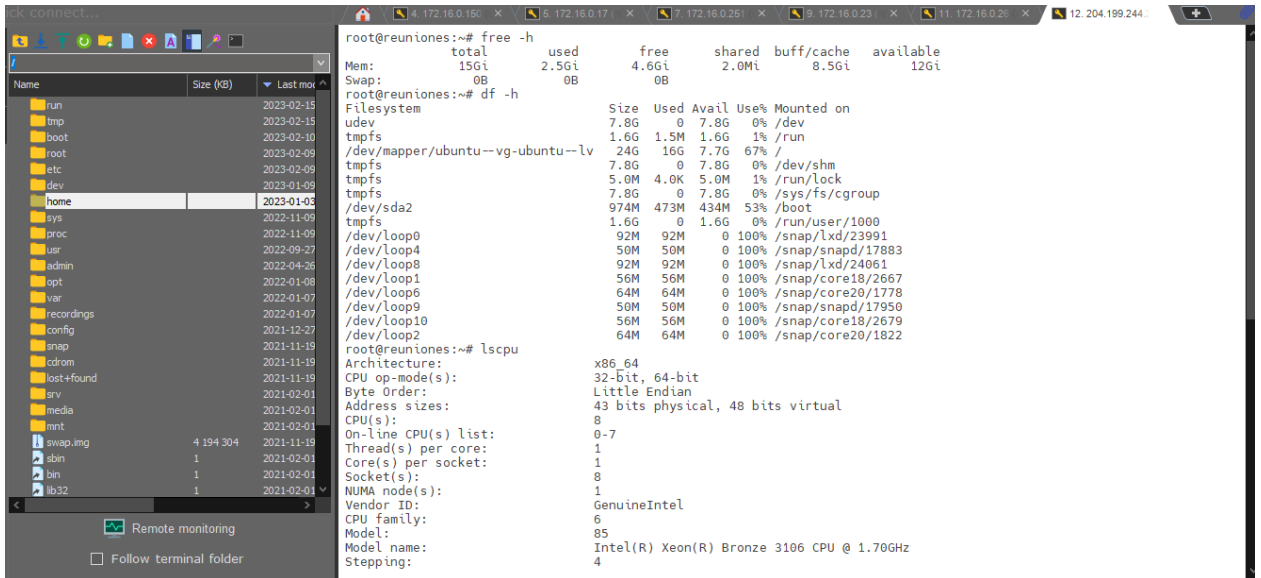


Figura 2.212 Ingreso al sistema virtualizado en UBUNTU mediante servicio SSH, VMWare en servidor real del ISTS.

En la figura 2.23 se muestra el acceso a la base de datos en MySQL, la base de datos db_usuarios_reuniones, la tabla prosody y sus columnas de información.

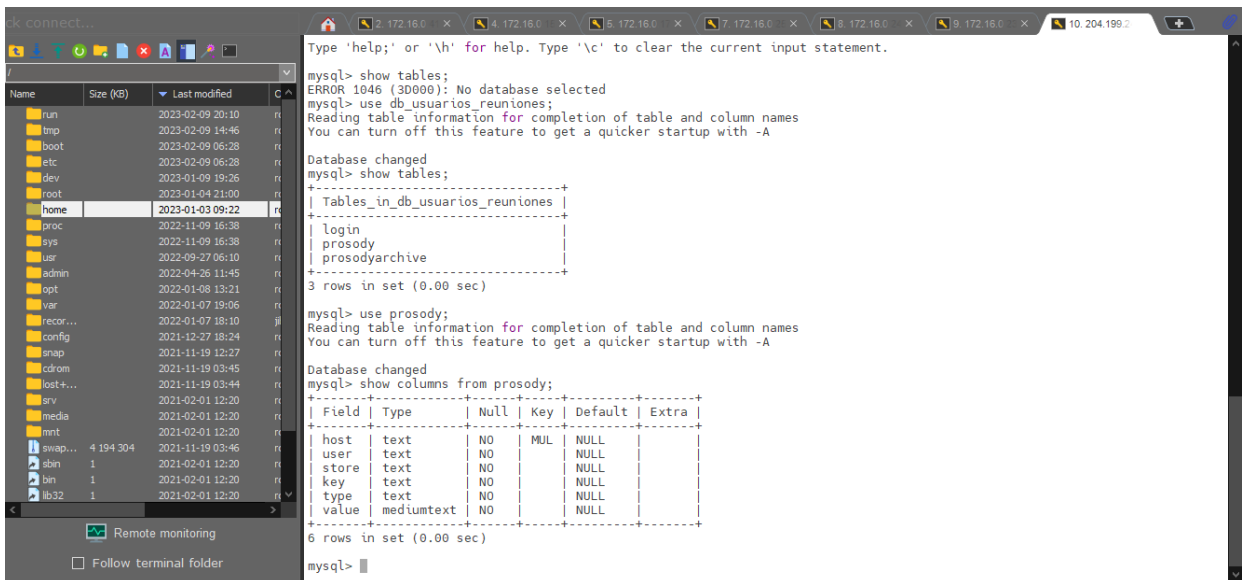


Figura 2.23. Pantalla MySQL para ingreso a la base de datos del sistema de videoconferencia.

La figura 2.24 muestra la infraestructura general de los servidores necesarios para el funcionamiento del prototipo, en el cual todos los servidores activos se conectan al servidor de autenticación para autenticarse con nombres de usuarios dedicados

a cada servidor activo para poder enlazarse al sistema de servidores donde trabajan en conjunto.



Figura 2.24. Servidores activos para el correcto funcionamiento del sistema de videoconferencia.

En la Tabla 2.3. se muestra la tabla prosody la cual se conecta con el servidor prosody para realizar la autenticación de usuarios, el modelo relacional se lo obtuvo con el programa Valentina Studio, cabe recalcar que este servidor de autenticación solo puede funcionar con la tabla prosody y con las siguientes variables y tipos de datos.

Variable	Tipo	Descripción
host	Text	Guarda al nombre de dominio del servidor de videoconferencias.
key	Text	Guarda el tipo de contraseña para usuarios se usa el tipo password.
store	Text	Guarda el tipo cuenta para usuarios se usa accounts.
type	Text	Guarda el tipo de dato para usuarios se utiliza string.
user	Text	Guarda el nombre de usuario.
value	MediumText	Guarda la contraseña del usuario.

Tabla 2.3. Atributos tabla prosody.

En la figura 2.25. se puede observar el Diagrama modelo relacional, donde reutilizamos el correo y la contraseña desde la tabla Usuario y lo enviamos a la base de datos del servidor de videoconferencia, a través de la tabla prosody la cual se comunica con el servidor de autenticación.

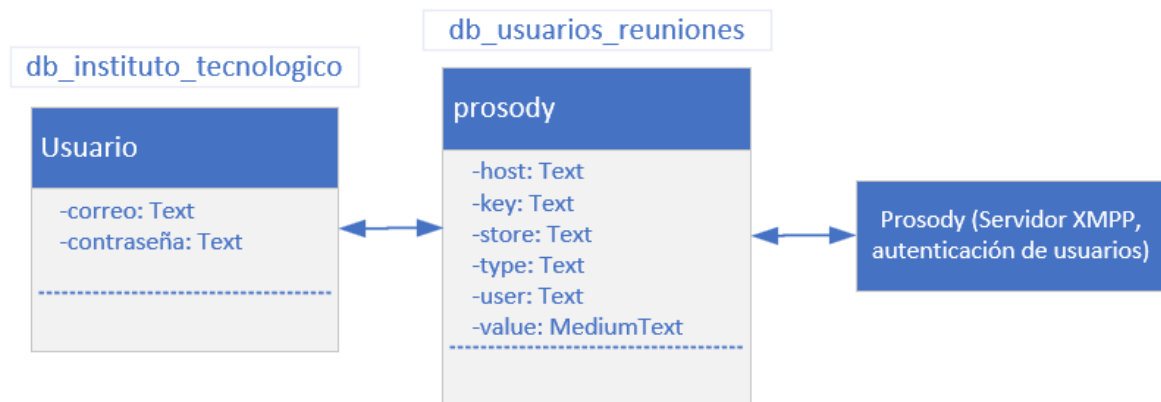


Figura 2.25. Diagrama modelo relacional.

En la figura 2.26., se puede observar las credenciales de los servidores de transmisión y grabación de videoconferencias, administrador de videoconferencias, comunicación multiusuario de video y los usuarios que pueden autenticarse con el prototipo de videoconferencia.

	host	user	store	key	type	value
1	auth.reuniones.ospecuador.net	focus	accounts	salt	string	3847fd54-c51a-45e1-8c55-85b448ee87d4
2	auth.reuniones.ospecuador.net	focus	accounts	iteration_count	number	4096
3	auth.reuniones.ospecuador.net	focus	accounts	stored_key	string	1dc5ffc9cc8aedc9054415c74d3990d4d18fd2f
4	auth.reuniones.ospecuador.net	focus	accounts	server_key	string	528a05823e142081157f85029e3863ad93ecd2a
5	auth.reuniones.ospecuador.net	jvb	accounts	salt	string	0b6e5b8d-4b0b-49d0-b642-026bc2db88a8
6	auth.reuniones.ospecuador.net	jvb	accounts	stored_key	string	ef0c5c87d9d609a70ac82b45268700e4d90e0c1
7	auth.reuniones.ospecuador.net	jvb	accounts	iteration_count	number	4096
8	auth.reuniones.ospecuador.net	jvb	accounts	server_key	string	2951705ce86ecadc71ec7f692ace5fbaf5de964
9	reuniones.ospecuador.net	admin	accounts	password	string	paradise
10	auth.reuniones.ospecuador.net	jibri	accounts	stored_key	string	7a2a62d3324156ac2ab5b5735bfe127f00ae0ac2
11	auth.reuniones.ospecuador.net	jibri	accounts	server_key	string	9b390eccb665c5f8eb653fb8b975d57b95c0bde
12	auth.reuniones.ospecuador.net	jibri	accounts	iteration_count	number	4096
13	auth.reuniones.ospecuador.net	jibri	accounts	salt	string	3e247156-d6f5-4195-80a9-84b943be84b5
14	recordecreuniones.ospecuador.net	recorder	accounts	iteration_count	number	4096
15	recordecreuniones.ospecuador.net	recorder	accounts	server_key	string	47f1b6389b70fa51197cce0aaa42981e9f7f825
16	recordecreuniones.ospecuador.net	recorder	accounts	salt	string	a6a643c9-a2ad-4d7c-8459-71f040f18e40
17	recordecreuniones.ospecuador.net	recorder	accounts	stored_key	string	3a8e84ec44458c5bac91209cb277eeb63e0fefa
18	reuniones.ospecuador.net	administrador	accounts	password	string	ists2022@admin
19	reuniones.ospecuador.net	usuario1	accounts	password	string	ists2022@user1
20	reuniones.ospecuador.net	usuario3	accounts	password	string	ists2022@user3
21	reuniones.ospecuador.net	usuario3	accounts	password	string	ists2022@user3
22	reuniones.ospecuador.net	usuario4	accounts	password	string	ists2022@user4
23	reuniones.ospecuador.net	usuario5	accounts	password	string	ists2022@user5
24	reuniones.ospecuador.net	usuario6	accounts	password	string	ists2022@user6
25	reuniones.ospecuador.net	usuario7	accounts	password	string	ists2022@user7
26	reuniones.ospecuador.net	usuario8	accounts	password	string	ists2022@user8

Figura 2.26. Contenido de la tabla prosody de la base de datos db_usuarios_reuniones.

En la figura 2.27. se muestra la asignación de IP pública al servidor local y apertura de puertos necesarios para el funcionamiento del servidor de videoconferencias en el firewall cisco meraki MX64.

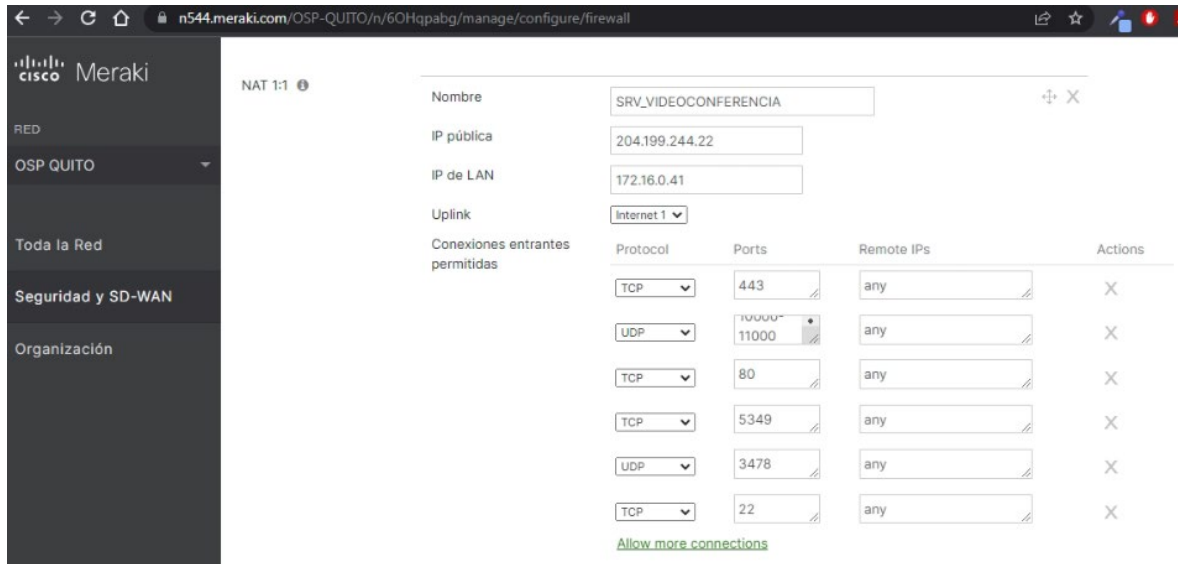


Figura 2.27. Asignación de IP pública al servidor local y apertura de puertos.

En la Tabla 2.4. se muestra los puertos necesarios que se deben habilitar para el correcto funcionamiento del servidor de videoconferencia.

Protocolo	Puerto	Descripción
TCP	443	Protocolo HTTPS
UDP	10000-11000	Tráfico multimedia
TCP	80	Protocolo HTTP
TCP	5349	Turn retransmitir tráfico
UDP	3478	Turn retransmitir tráfico
TCP	22	Protocolo SSH

Tabla 2.4. Puertos abiertos en el servidor de videoconferencia

2.3 PRUEBAS PRELIMINARES

Una vez terminado el proceso de diseño e implementación se procede a realizar las pruebas preliminares como se estableció dentro del tablero KANBAN, la figura 2.28 muestra la actualización de este.

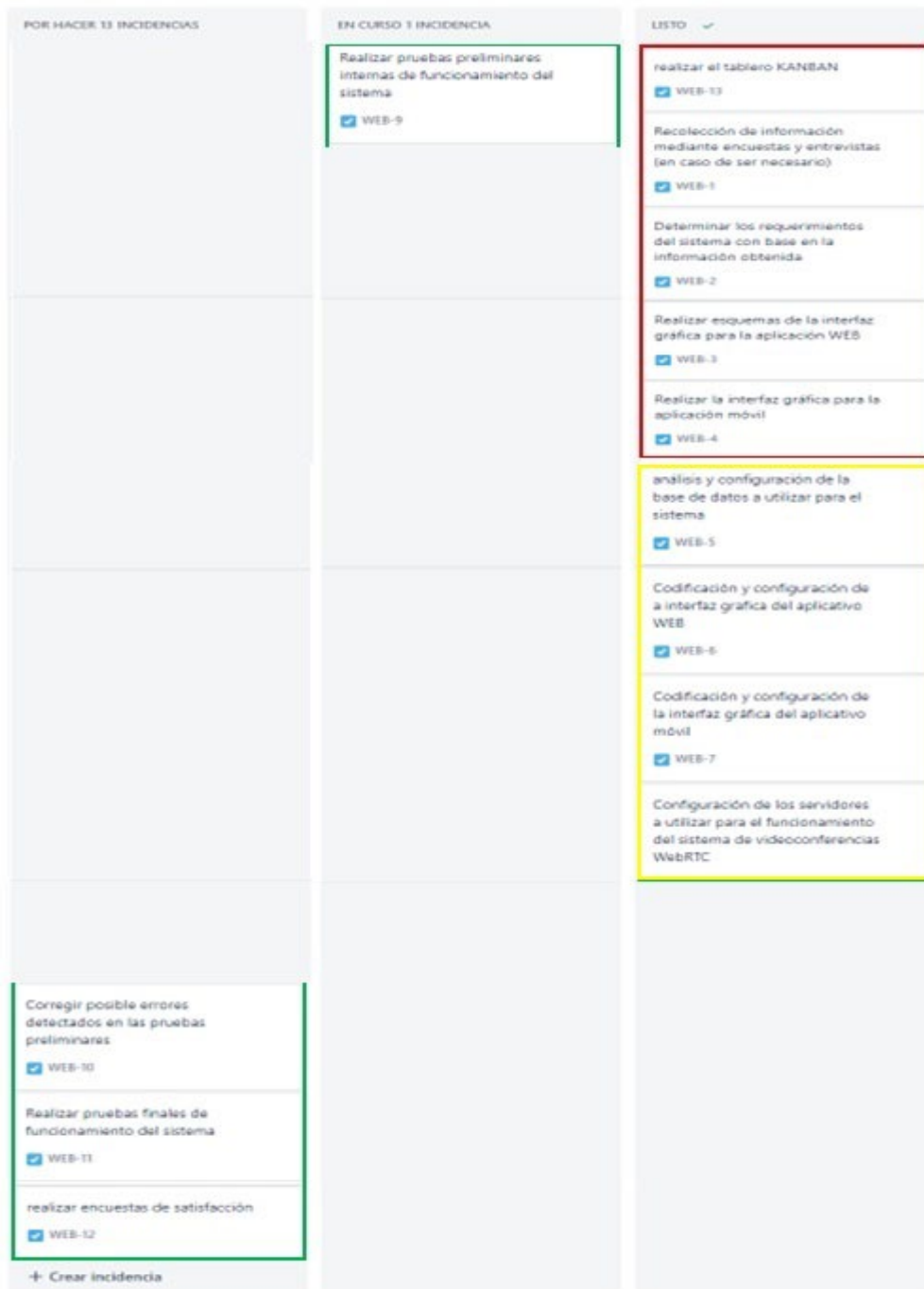


Figura 2.28. Tablero KANBAN actualizado.

Para la realización de las pruebas preliminares del sistema, se procede a ingresar en el aplicativo a través de la dirección de dominio desde un computador o un dispositivo móvil, mismos que realizan las funciones de moderador y usuario, respectivamente, cabe mencionar que ambos equipos se encuentran conectados fuera de la red del ISTS.

Como se mencionó la aplicación se encuentra organizada por módulos; registro, autenticación, comunicación de usuarios, mensajería instantánea, compartición de pantalla, grabación de videoconferencia, transmisión en vivo de YouTube, módulo de ayuda y módulo notificaciones.

Para la implementación de la aplicación se tomará como base el módulo de Autenticación debido a que los demás se implementan de manera similar.

Dentro de este se tiene algunas carpetas, sin embargo, la carpeta *components* es la principal en ella se encuentra la codificación de parte visual (vista). Cada vista tiene propiedades denominadas *PROPS* y estados denominados *STATE*, variables que influyen en el resultado del render, método encargado de la renderización de la página JavaScript hacia HTML.

- **State:** Se administra dentro del componente.
- **Props:** Son los parámetros que se pasan al componente.

Cabe mencionar que la implementación de los módulos tiene su base en el gestor de estados para aplicaciones que permite el almacenamiento de información, "REDUX". Este gestor está conformado por *Actions* y *Reduce*.

- **Actions:** Encargado de enviar datos de la aplicación hacia el almacenamiento (local Storage).
- **Reduce:** método de respuesta a una acción (action) en conjunto con un estado que se ejecutan como parámetros de entrada, es decir encargado de realizar una acción específica proporcionada por *REDUX*.

Para lograr la implementación de este módulo, se tienen varios archivos y códigos de programación, sin embargo, se detallarán los más importantes.

LOGINDIALOG.JS

Este archivo representa la interfaz gráfica que se presentará al usuario final para que realice la autenticación dentro del sistema y se le otorguen los privilegios. La vista (interfaz) es una clase que se encuentra dentro de la carpeta *component*, mismo que recibe los datos de tipo *Props* y *State*, como se observa en la línea de programación 52. Cada clase tiene su constructor que permite inicializar variables que se requieran, por ejemplo, *state* debe ser inicializado con este, se puede

observar en las líneas 57 a 60 del código de programación presentado en la figura 2.29.

```
52 class LoginDialog extends Component<Props, State> {
53
54   constructor(props: Props) {
55     super(props);
56
57     this.state = {
58       username: '',
59       password: '',
60       loginStarted: false
61     };
62
63     this._onCancelLogin = this._onCancelLogin.bind(this);
64     this._onLogin = this._onLogin.bind(this);
65     this._onChange = this._onChange.bind(this);
66   }
67
68   _onCancelLogin: () => void;
69
70   onCancelLogin() {
71     const { dispatch } = this.props;
72
73     dispatch(cancelLogin());
74   }
}
```

Figura 2.29. Código de programación para autenticación de usuario en el sistema.

Todas las vistas poseen un método `render` dentro de la clase, el cual se indica en la figura 2.30. El método `render` toma las propiedades *Props* (línea 241) y los estados *state* (línea 244) para construir la página. Además de especificar estas variables, aquí se construye toda la parte visual de la página. como son los inputs, botones, textos y más. En la línea 261 se crea un campo de texto el cual le permitirá a la persona ingresar su usuario, este posee propiedades que son utilizadas de acuerdo a los requerimientos del usuario, por ejemplo, *label* permite colocarle una etiqueta a este texto, *value* permite asignar un valor a la variable con la cual se va a trabajar en la implementación de métodos. En la línea 272 se crea otro campo de texto para que el usuario ingrese su contraseña, *TextField* es un componente propio de *react*.

```
239 → render() {
240     const {
241       _connecting: connecting,
242       t
243     } = this.props;
244     const { password, loginStarted, username } = this.state;
245
246     return (
247       <Dialog
248         disableBlanketClickDismiss = { true }
249         hideCloseIconButton = { true }
250         okDisabled = {
251           connecting
252           || loginStarted
253           || !password
254           || !username
255         }
256         okKey = { t('dialog.login') }
257         onCancel = { this._onCancelLogin }
258         onSubmit = { this._onLogin }
259         titleKey = { t('dialog.authenticationRequired') }
260         width = { 'small' } >
261         <TextField
262           autoFocus = { true }
263           className = 'input-control'
264           compact = { false }
265           label = { t('dialog.user') }
266           name = 'username'
267           onChange = { this._onChange }
268           placeholder = { t('dialog.userIdentifier') }
269           shouldFitContainer = { true }
270           type = 'text'
271           value = { username } />
272         <TextField
273           className = 'input-control'
274           compact = { false }
275           label = { t('dialog.userPassword') }
276           name = 'password'
277           onChange = { this._onChange }
278           placeholder = { t('dialog.password') }
279           shouldFitContainer = { true }
280           type = 'password'
281           value = { password } />
282         { this.renderMessage() }
283       </Dialog>
284     );
285 }
```

Figura 2.30. Programación de renderización de la página, ingreso de usuario y contraseña del usuario.

Las propiedades y estados se especifican al inicio del código de la clase *LoginDialog*, figura 2.31, y son creados mediante la palabra reservada *type* seguida de la palabra *Props* o *state* para definir propiedades o estados correspondientemente, a continuación, se indican cada uno de ellos.

```

19  type Props = {
20
21      _conference: Object,
22
23      _configHosts: Object,
24
25      _connecting: boolean,
26
27      _error: Object,
28
29      _progress: number,
30
31      dispatch: Dispatch<any>,
32
33      onSuccess: Function,
34
35      roomName: string,
36
37      t: Function
38  }

```

(a)

```

41  type State = {
42
43
44      password: string,
45
46      username: string,
47
48      loginStarted: boolean
49  }
50

```

(b)

Figura 2.31. Programación de *Props* (a) y *State* (b).

La clase declara los métodos a utilizar y luego realiza la implementación del mismo, previo a esto todos los métodos que se declaren en la clase deben ser instanciados mediante la función *bind* la cual permite especificar este sea propio de la clase, esto se observa en la figura 2.32. En la línea 64 se instancia el método *_onCancelLogin*, el mismo que se declara en la línea 68, este no recibe parámetros de entrada y es sin retorno. En la línea 70 se realiza la implementación y dentro de este se realiza toda la lógica necesaria para el funcionamiento de la pantalla. Para manejar los cambios de estado que requiere las pantallas, *REDUX* hace uso de *Dispatch*, este permite actualizar las variables almacenadas en el *storage* e indicar estos valores al usuario.

```

52 class LoginDialog extends Component<Props, State> {
53
54     constructor(props: Props) {
55         super(props);
56
57         this.state = {
58             username: '',
59             password: '',
60             loginStarted: false
61         };
62
63         this._onCancelLogin = this._onCancelLogin.bind(this);
64         this._onLogin = this._onLogin.bind(this);
65         this._onChange = this._onChange.bind(this);
66     }
67
68     _onCancelLogin: () => void;
69
70     _onCancelLogin() {
71         const { dispatch } = this.props;
72
73         dispatch(cancelLogin());
74     }

```

Figura 2.32. Programación de métodos a utilizar.

```

76     _onLogin: () => void;
77
78     onLogin() {
79         const {
80             _conference: conference,
81             _configHosts: configHosts,
82             roomName,
83             onSuccess,
84             dispatch
85         } = this.props;
86         const { password, username } = this.state;
87         const jid = toJid(username, configHosts);
88
89         if (conference) {
90             dispatch(authenticateAndUpgradeRole(jid, password, conference));
91         } else {
92             this.setState({
93                 loginStarted: true
94             });
95
96             connect(jid, password, roomName)
97                 .then(connection => {
98                     onSuccess && onSuccess(connection);
99                 })
100                .catch(() => {
101                    this.setState({
102                        loginStarted: false
103                    });
104                });
105         }
106     }
107

```

Figura 2.33. Programación del método _onlogin.

En la figura 2.33 se muestra la implementación del método `_onLogin`, donde desde línea 79 a la 85 toma los *Props* y en la línea 86 recurre a los *state* que requiere para la implementación. A partir de la línea 89 se tiene la lógica para el funcionamiento de la autenticación del usuario.

3. RESULTADOS Y DISCUSIÓN

Con el objetivo de realizar la validación del sistema propuesto, en este apartado se presentarán los resultados de las pruebas de funcionalidad relevantes. Previa a la presentación de estas, se procede a realizar una actualización del tablero KANBAN, figura 3.1, para determinar el avance de las actividades planteadas al inicio del proyecto.

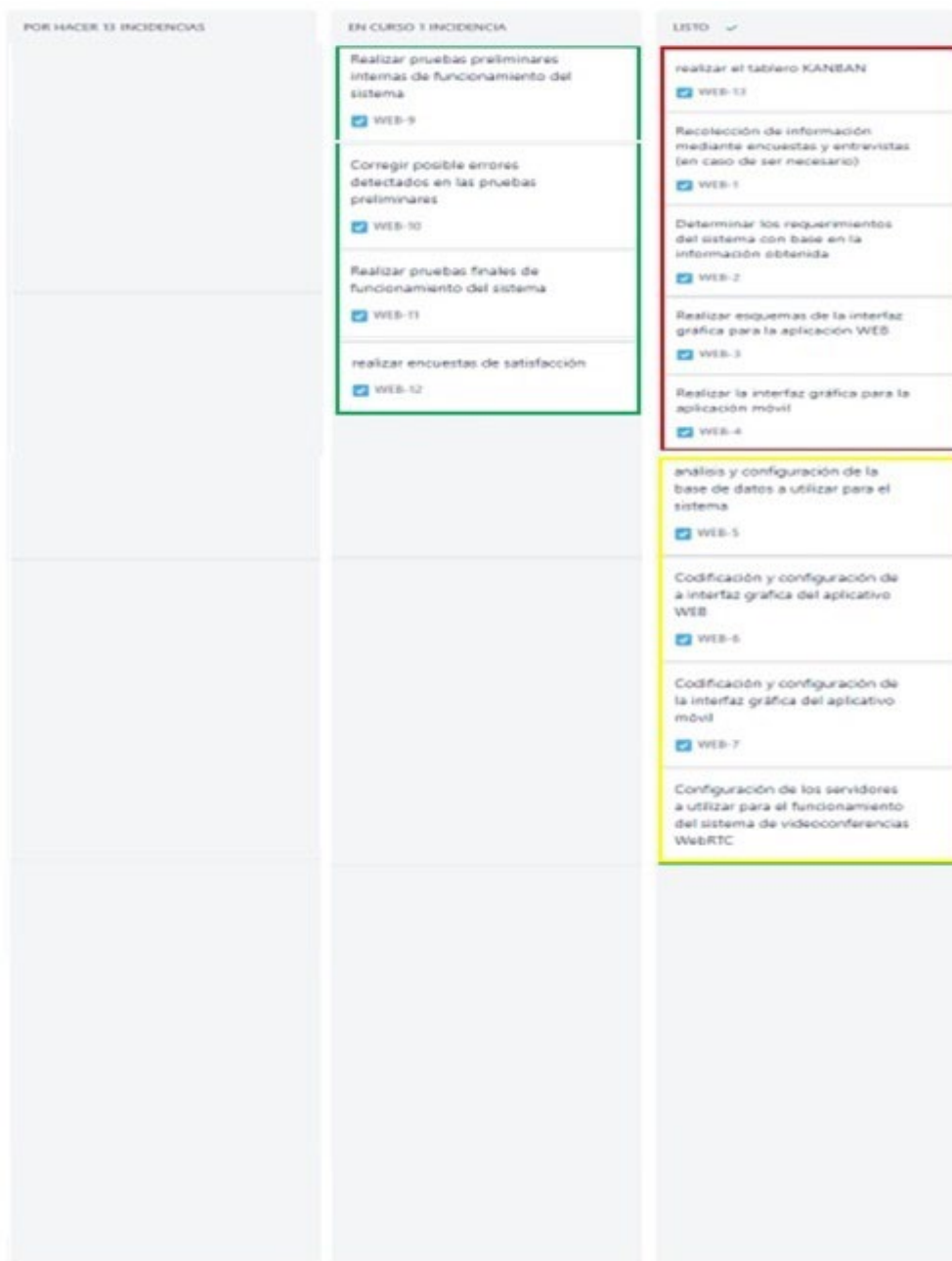


Figura 3.22. Tablero KANBAN actualizado.

Las pruebas del sistema se efectuaron en un computador dentro del navegador Google Chrome y en dispositivos móviles con Android 12.0. Las pruebas realizadas están diseñadas para medir la funcionabilidad en cada uno de los dispositivos de manera individual, midiendo la comunicación entre ellos y los servicios de los servidores virtualizados.

3.1 AUTENTICACIÓN

En la figura 3.2 se presenta la creación de una sala virtual denominada “Auditorio 1”, misma que una vez iniciada, pide que se autentique y en caso de no ser un moderador, esperar a que este ingrese para iniciar la conferencia, mientras en la figura 3.3 se muestra los campos para ingresar con privilegios de moderador. Si las credenciales ingresadas en el formulario son erróneas se presentará un mensaje de error, figura 3.4.

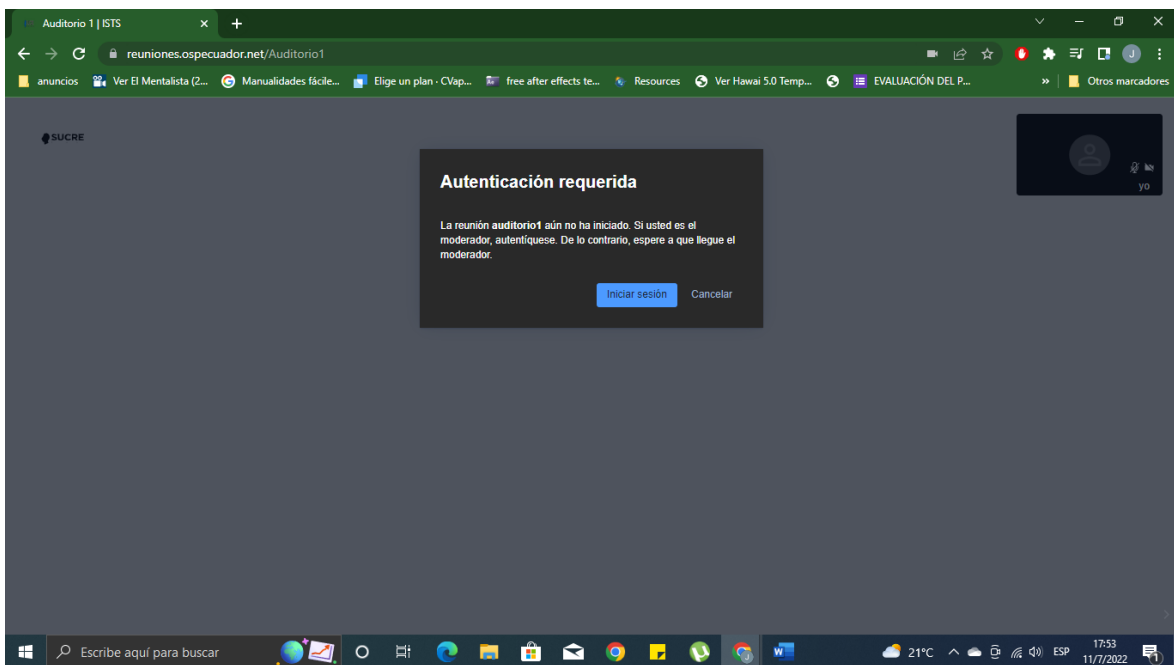


Figura 3.23. Creación de Sala conferencia, para ejemplo "Auditorio 1".

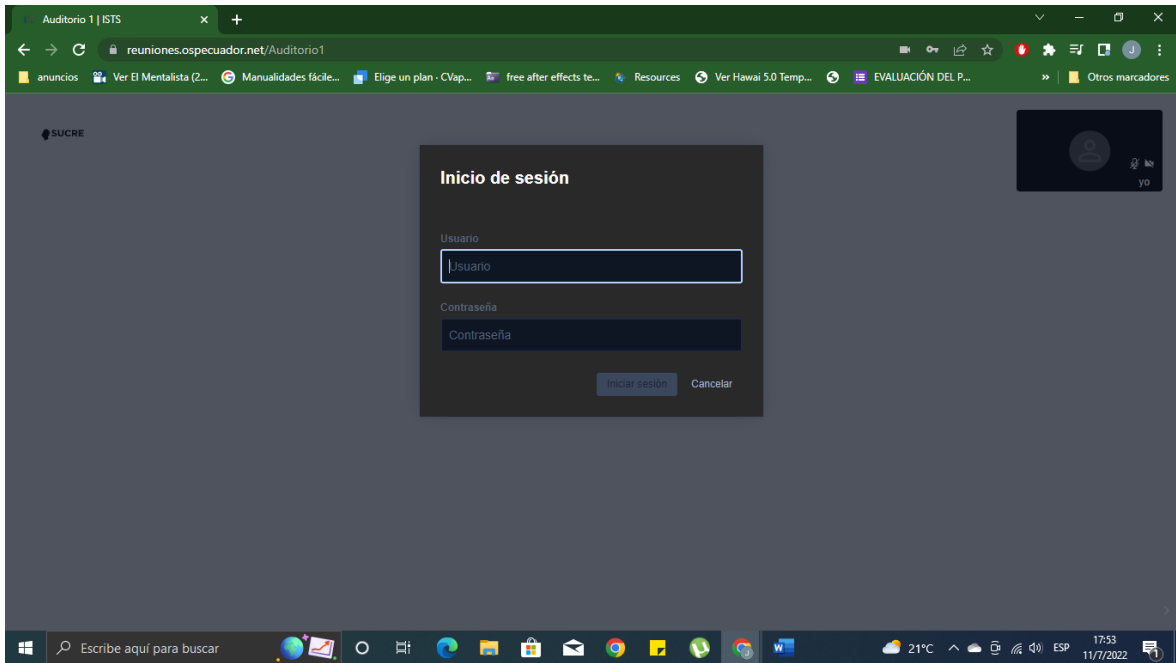


Figura 3.24. Formulario de ingreso para moderador con usuario y clave asignada.

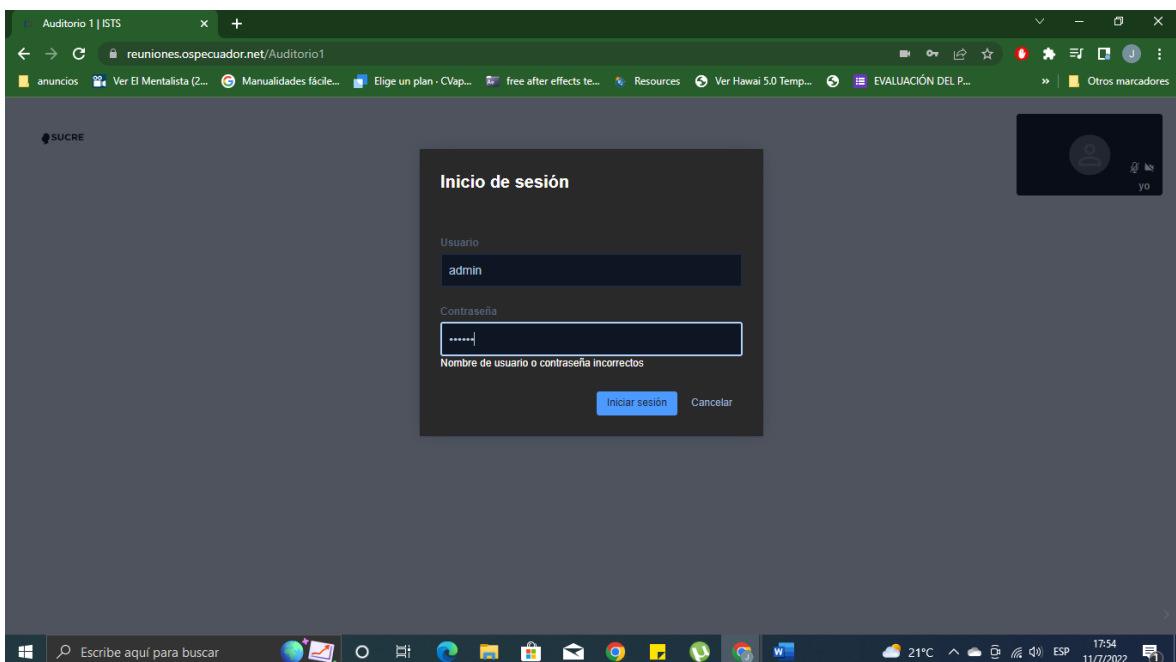


Figura 3.25. Error en la autenticación del moderador.

Cuando las credenciales ingresadas son correctas se inicia el modo moderador y se activa la sala creada, para el caso de ejemplo “Auditorio1”. La figura 3.5 muestra la sala activa con todas las opciones habilitadas.

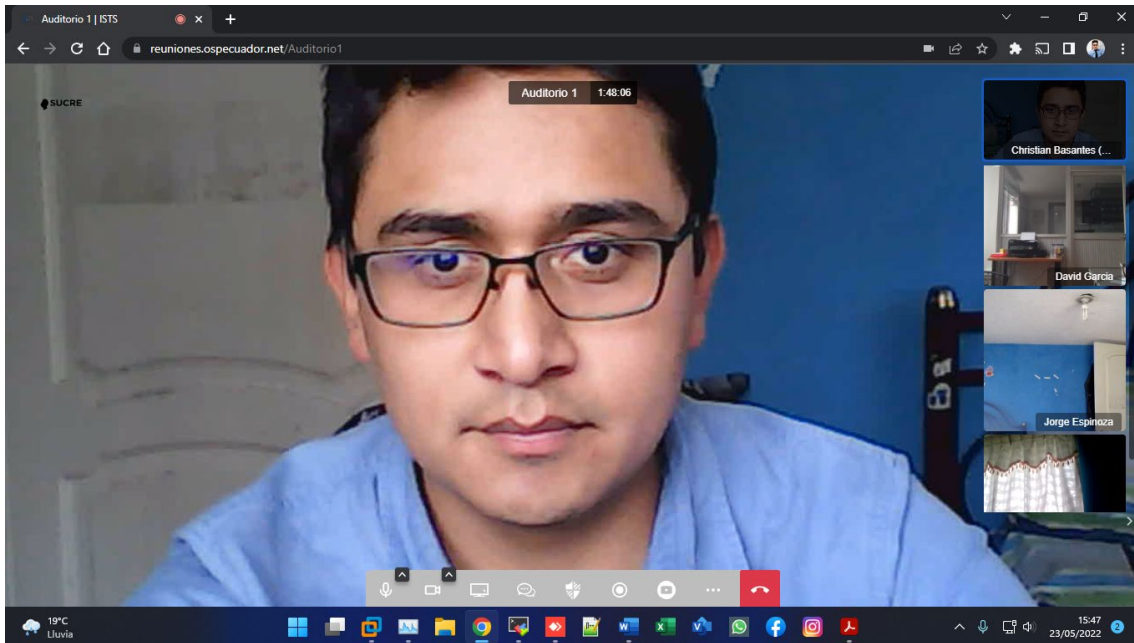


Figura 3.26. Salón virtual "Auditorio 1" activo en modo moderador.

Una vez habilitado el salón virtual con la entrada del moderador, los participantes pueden ingresar a la sesión únicamente escribiendo el nombre de la sesión activa a la que requieran unirse, como se muestra en la figura 3.6.

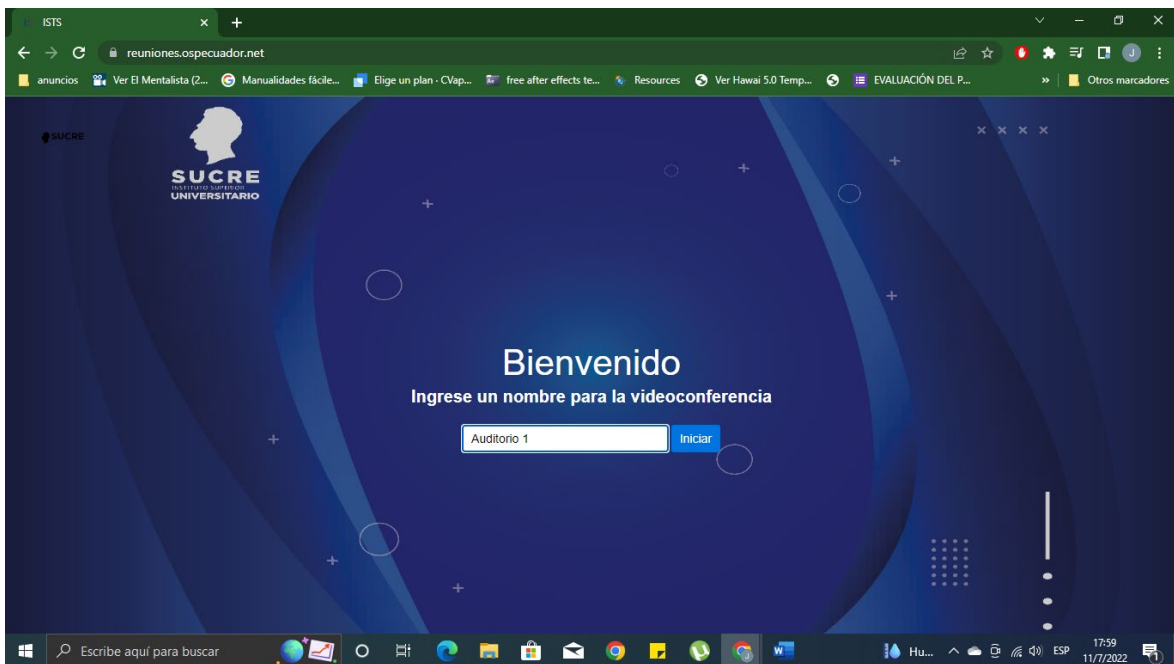


Figura 3.27. Ingreso de participantes a la sesión activa.

3.2 SERVICIO DE MENSAJERÍA INSTANTÁNEA (CHAT)

Una vez iniciada la sesión, se procede a realizar la prueba con cuatro usuarios a la vez como se muestra en la figura 3.7. durante esta sesión se ha permitido que cualquiera que conozca el nombre se pueda unir sin ningún tipo de restricción. De la misma manera el sistema tiene incluido un sistema que permite el envío de paquetes de datos de manera instantánea para la comunicación entre los usuarios de manera individual o colectiva, mediante este proceso se puede realizar el envío de mensajes simples de texto o archivos de texto. Durante las pruebas se realizaron envíos de mensajes mediante el sistema de mensajería incluida dentro del prototipo, para la ejecución de este proceso no se encontró novedad alguna.

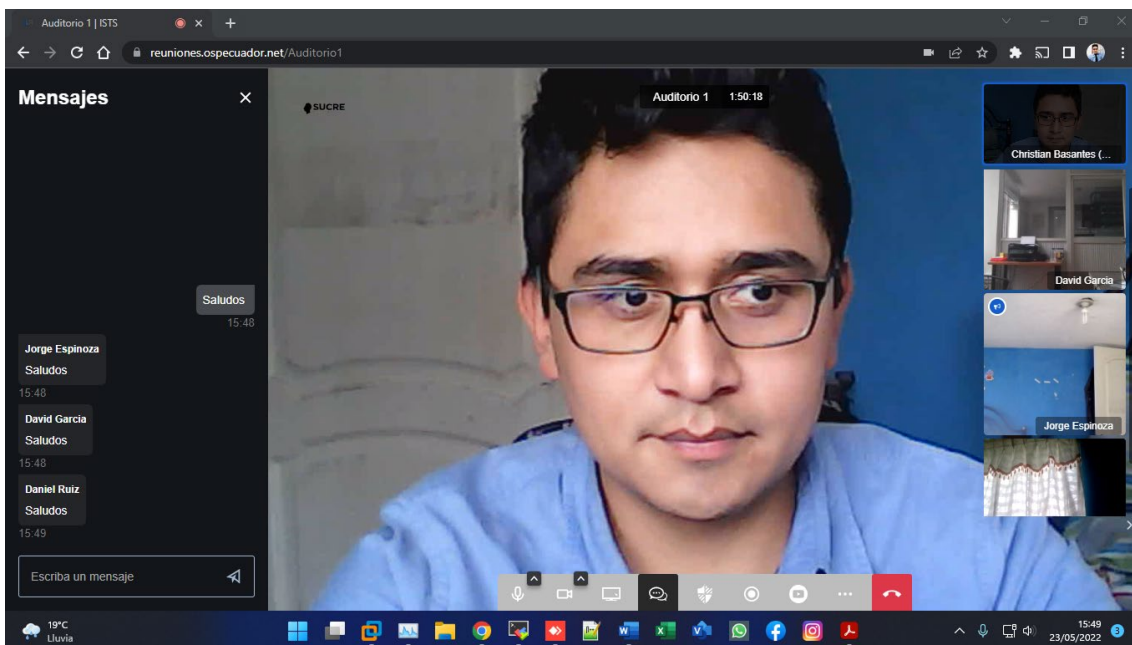


Figura 3.28. Reunión con cuatro (4) participantes y envío de mensajes mediante el sistema de chat del prototipo.

3.3 COMPARTICIÓN DE PANTALLA

Otra función importante y necesaria para este tipo de sistemas, es la opción de compartir pantalla, figura 3.8, para poder revisar documentos o información relevante que se desee presentar a los usuarios de la sesión. En la figura 3.9 se muestra la prueba realizada para la compartición de pantalla.

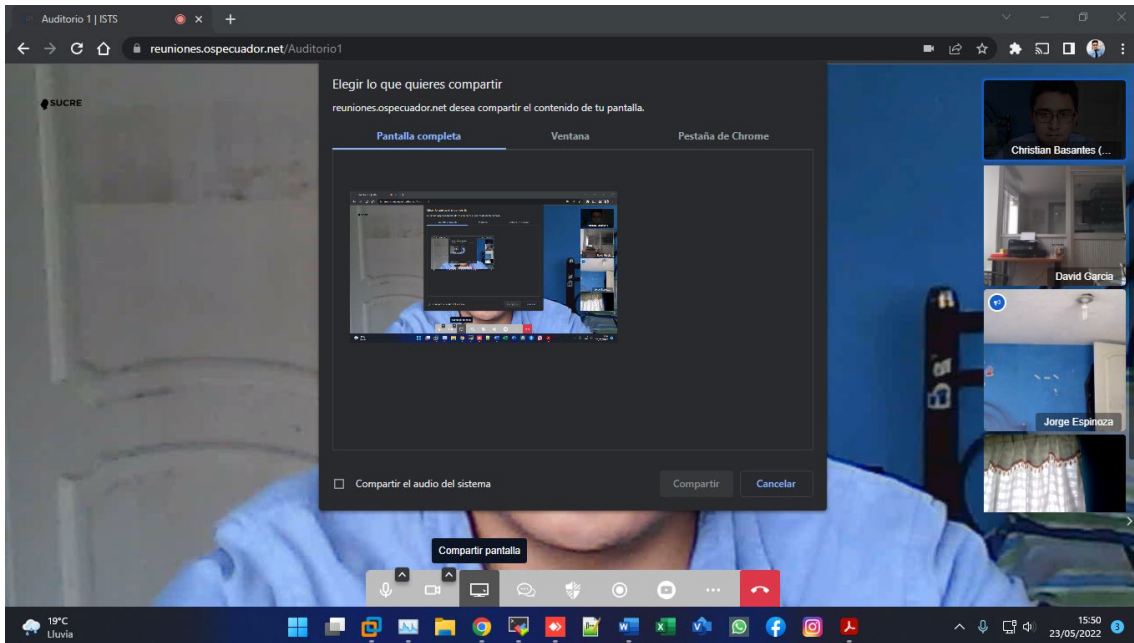


Figura 3.29. Activación de modo compartición de pantalla.

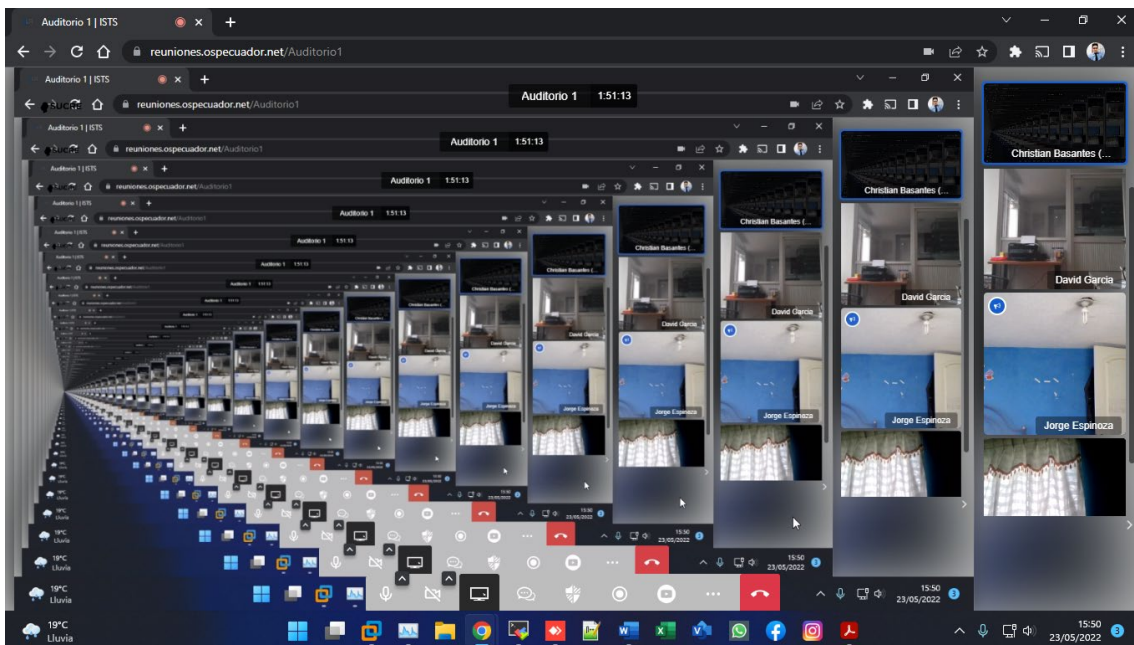


Figura 3.30. Presentación de pantalla compartida.

3.4 GRABACIÓN DE REUNIÓN

El usuario registrado como moderador tiene la opción de realizar la grabación de la reunión y posterior a ello esta se guardará en el servidor (local storage), mencionado en el capítulo anterior. De esta manera es capaz de compartir el

archivo con los usuarios de la sesión, en las figuras 3.10 y 3.11 se presenta la prueba realizada para la grabación de la sesión y el resultado de la misma.

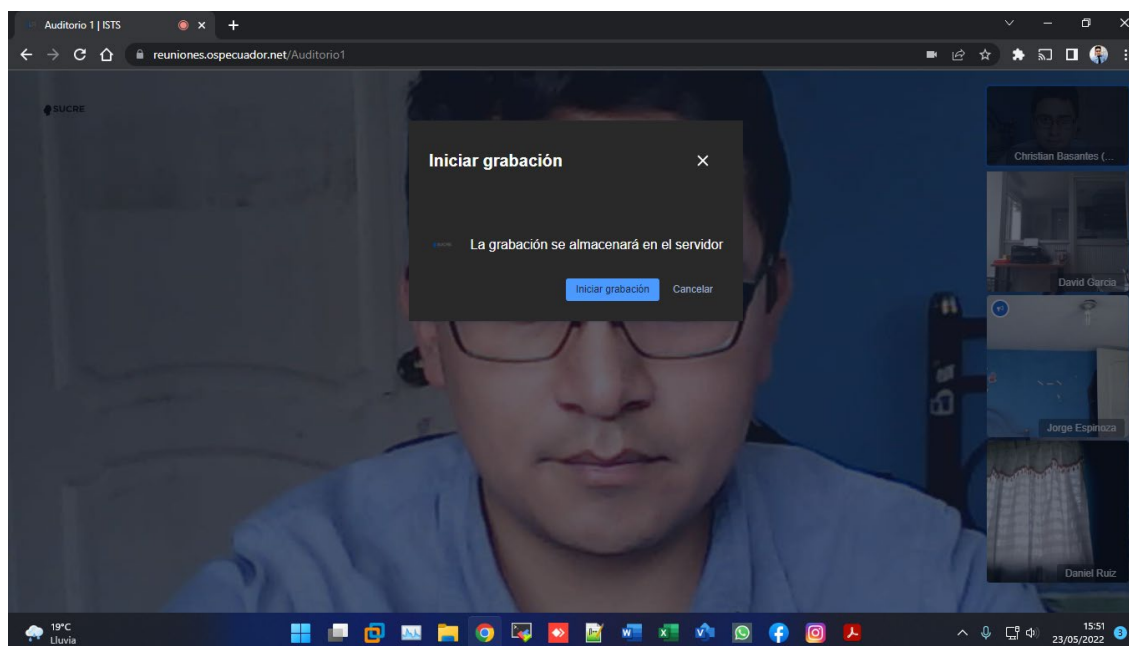


Figura 3.31. Activación de grabación de sesión.

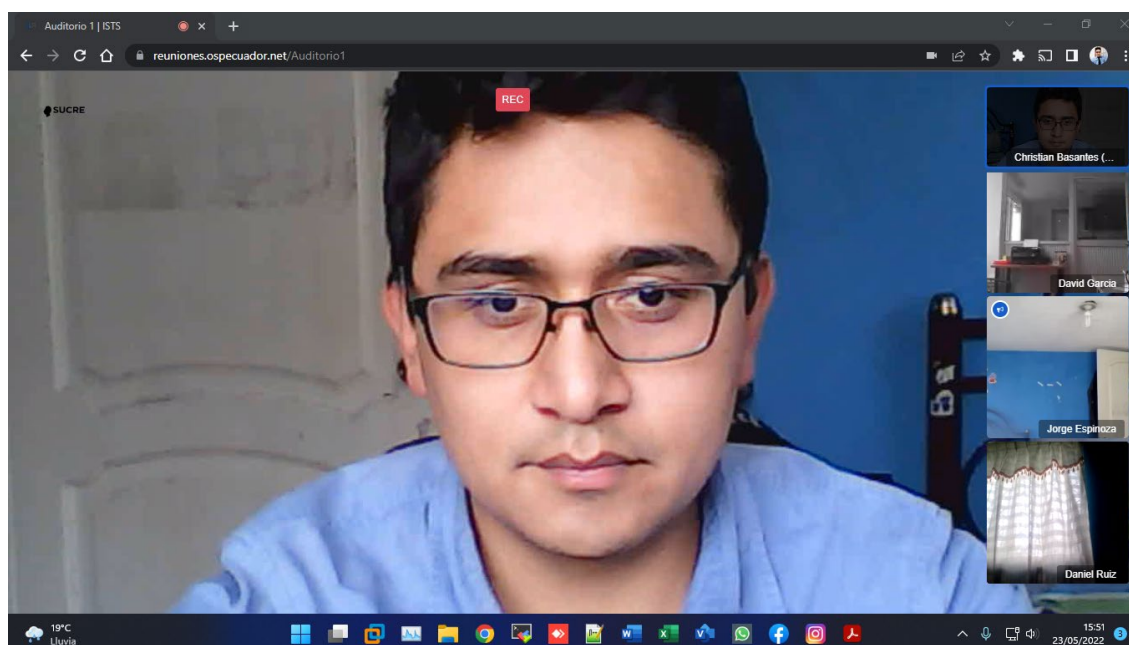


Figura 3.32. Grabación de Sesión, ejecutada por el moderador.

3.5 TRANSMISIÓN EN YOUTUBE

A diferencia de los otros prototipos utilizados para el mismo propósito, está tiene la particularidad de poder realizar una transmisión en vivo en el prototipo de YouTube.

Esta característica está habilitada únicamente para el moderador de la sala. en las figuras 3.12, 3.13, 3.14 y 3.15 se muestran las pruebas de la transmisión de la sesión “Auditorio 1”. Cabe recalcar que para realizar este tipo de transmisión se debe tener acceso a un código de transmisión emitido por YouTube desde su prototipo en la web, en el apartado “transmisión” desde un canal existente.

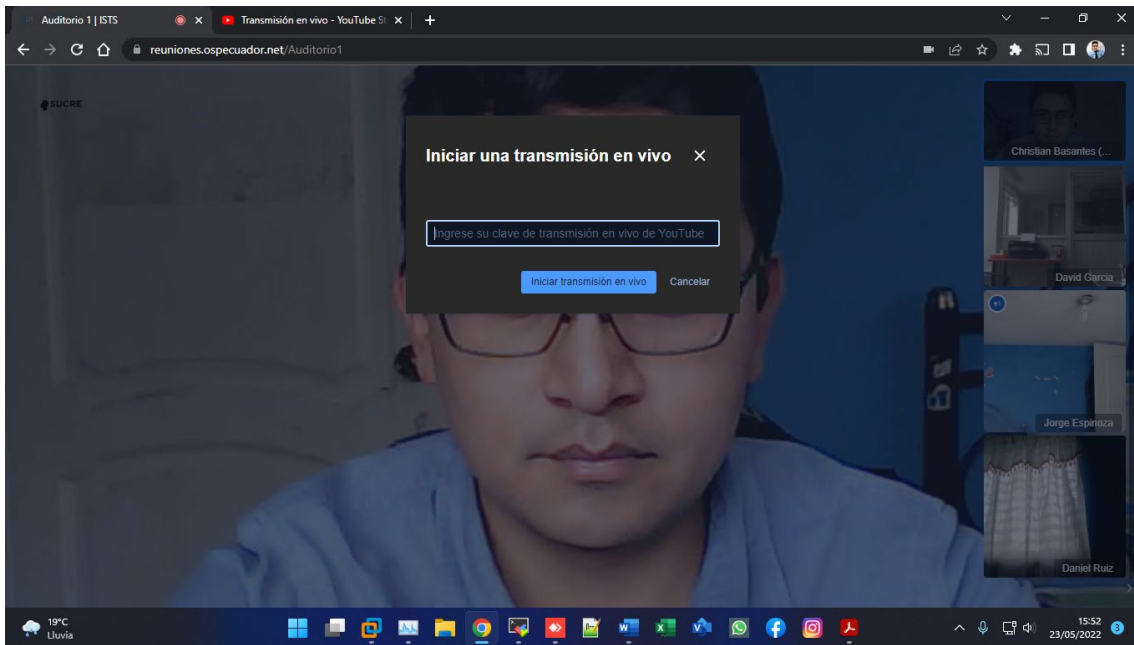


Figura 3.33. Solicitud de transmisión por YouTube.

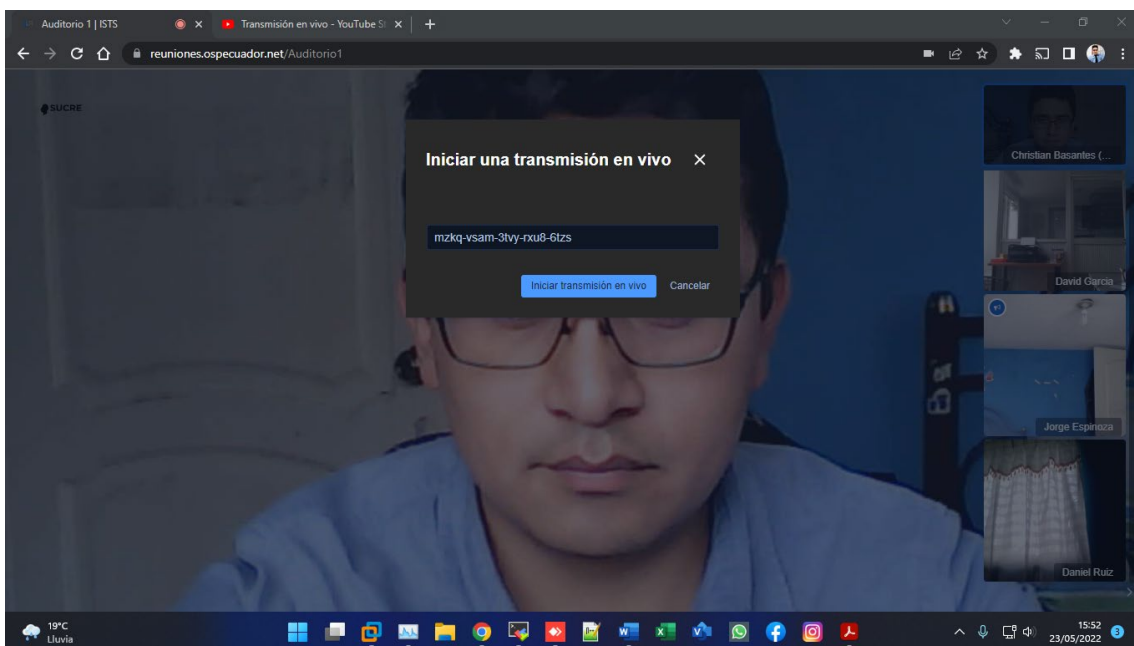


Figura 3.34. Activación de transmisión con el código de transmisión obtenido desde YouTube.

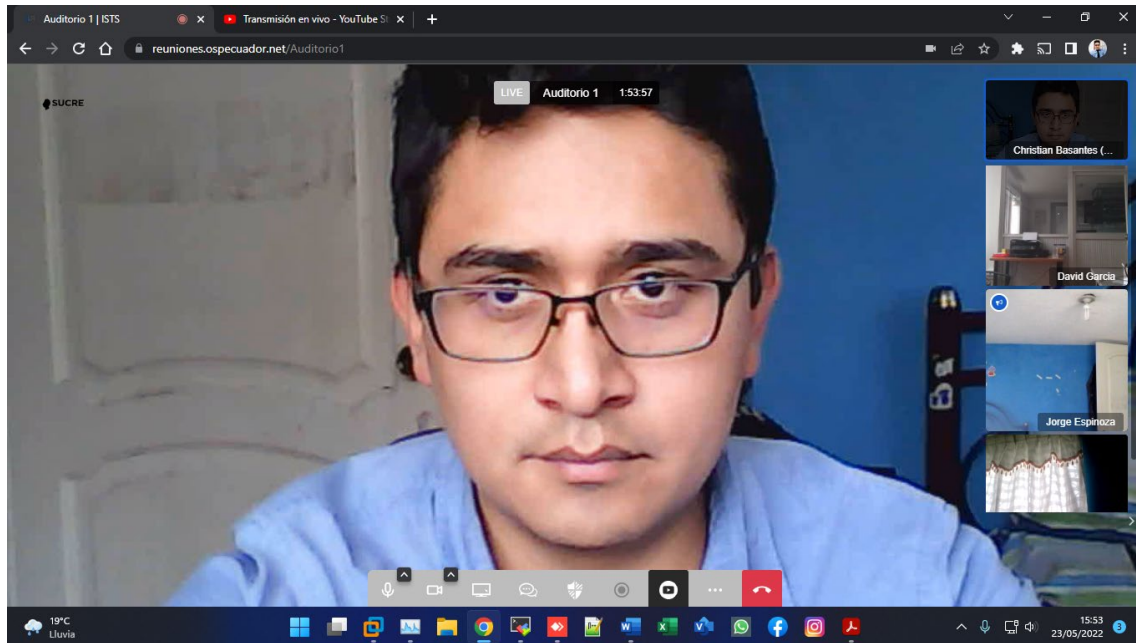


Figura 3.35. Transmisión por YouTube activado. se presenta la notificación en la parte superior junto al nombre de la sala.

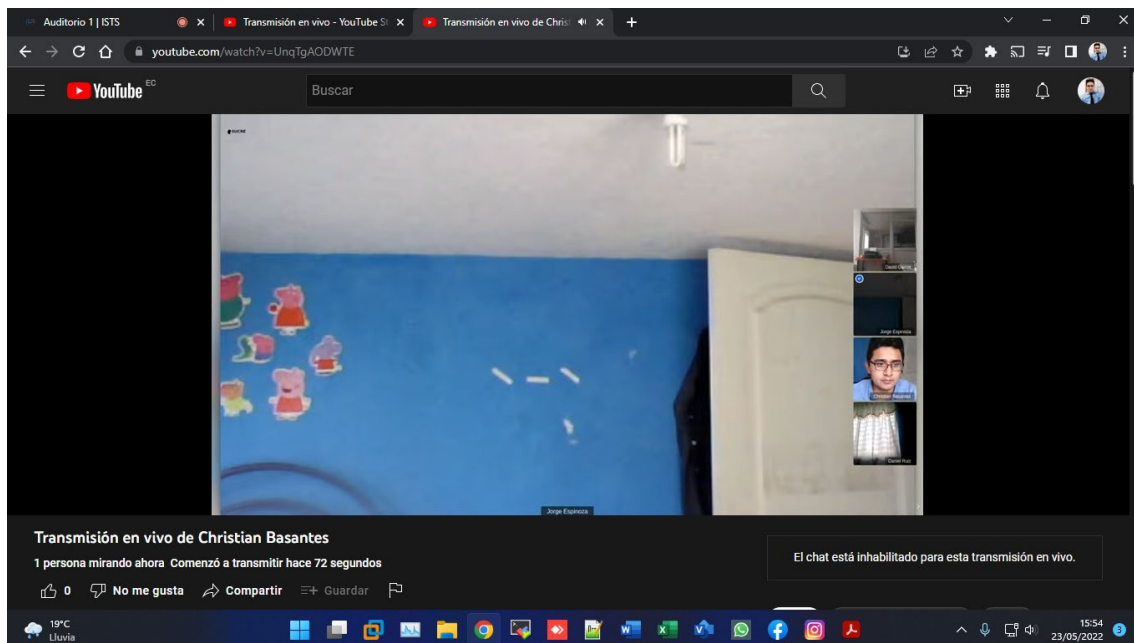


Figura 3.36. Transmisión de la sesión en el prototipo YouTube.

3.6 ASIGNACIÓN DE SEGURIDAD A LA SESIÓN

De la misma forma una vez iniciada la sesión el moderador tiene la opción de registrar una contraseña para la reunión, donde una vez que se ejecute esta operación los nuevos usuarios deberán conocer este parámetro para ingresar. A

continuación, en la figura 3.16 se presenta la prueba de asignación de contraseña para la sesión “Auditorio 1”.

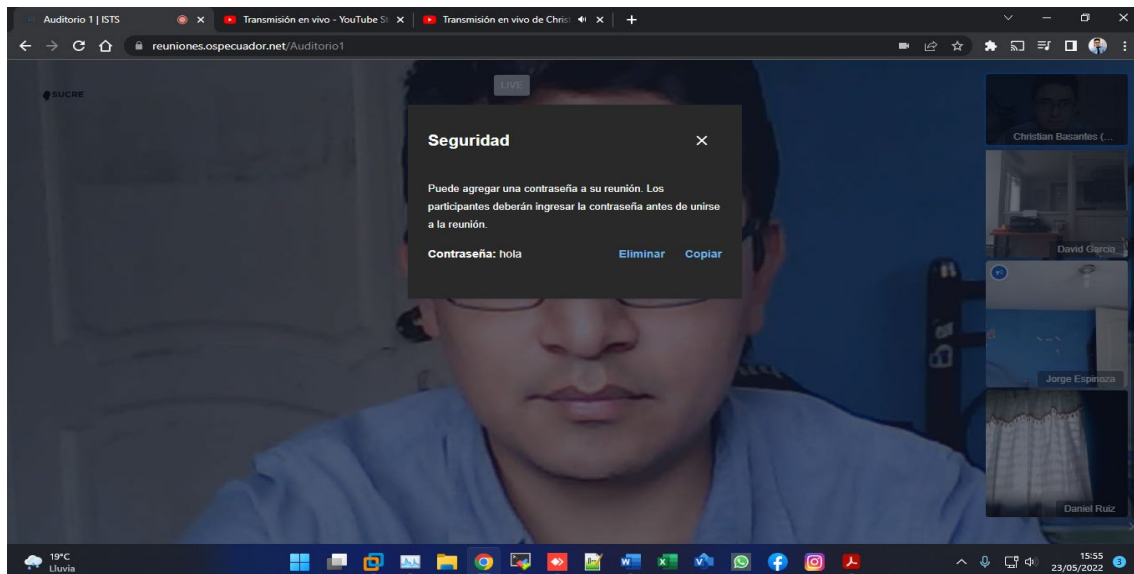


Figura 3.37. Configuración de seguridad para ingreso a la sesión "Auditorio 1".

3.7 NOTIFICACIONES AL ADMINISTRADOR

Las notificaciones enviadas al administrador son de tipo “correo” se abre una ventana auxiliar donde se colocan por defecto el emisor (usuario que envía la notificación), y el receptor (Administrador de la sesión), con la opción de colocar uno o más receptores en el cuadro “CC” (con copia). En la figura 3.17 se muestra la ventana auxiliar mostrada en la realización de las pruebas.

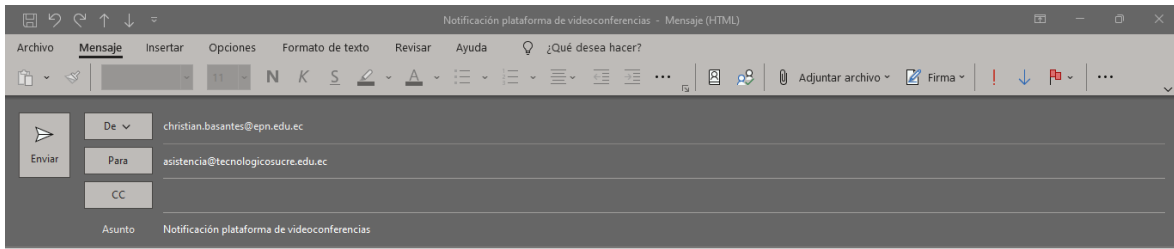


Figura 3.38. Ventana auxiliar de envío de notificaciones.

3.8 PÁGINA DE AYUDA

Dentro de la aplicación existe un módulo de ayuda, al cual se accede mediante un sub menú que se muestra en la figura 3.18 de la prueba realizada. Dentro de esta ayuda se presenta una guía con la explicación del funcionamiento básico del prototipo, como se muestra en la figura 3.19.

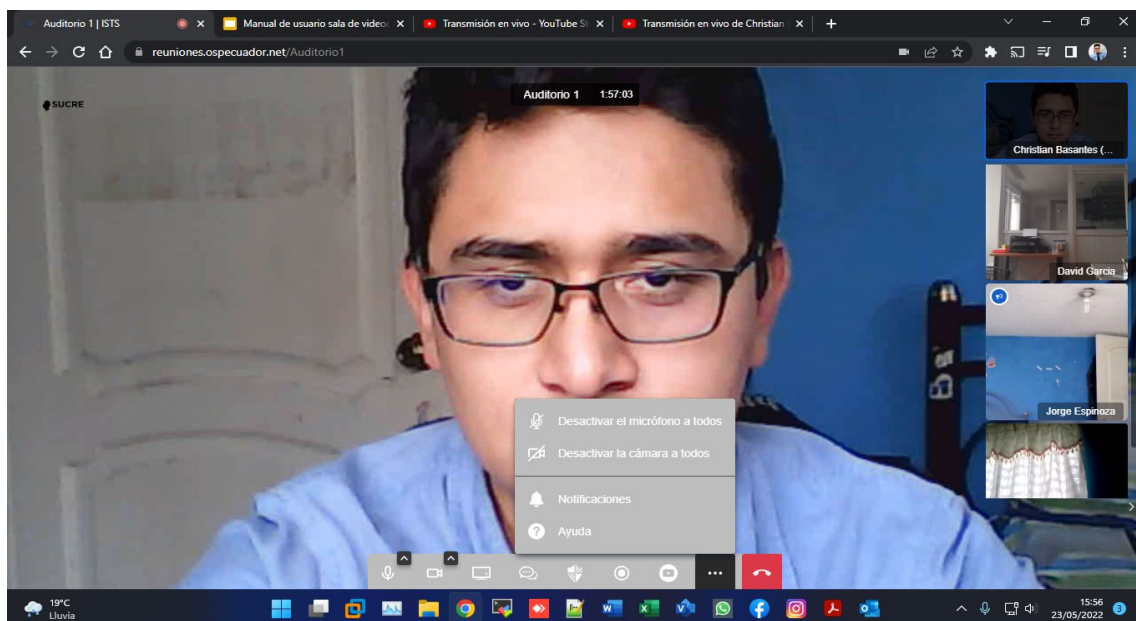


Figura 3.39. Sub menú del sistema para ingreso al módulo de ayuda del prototipo.

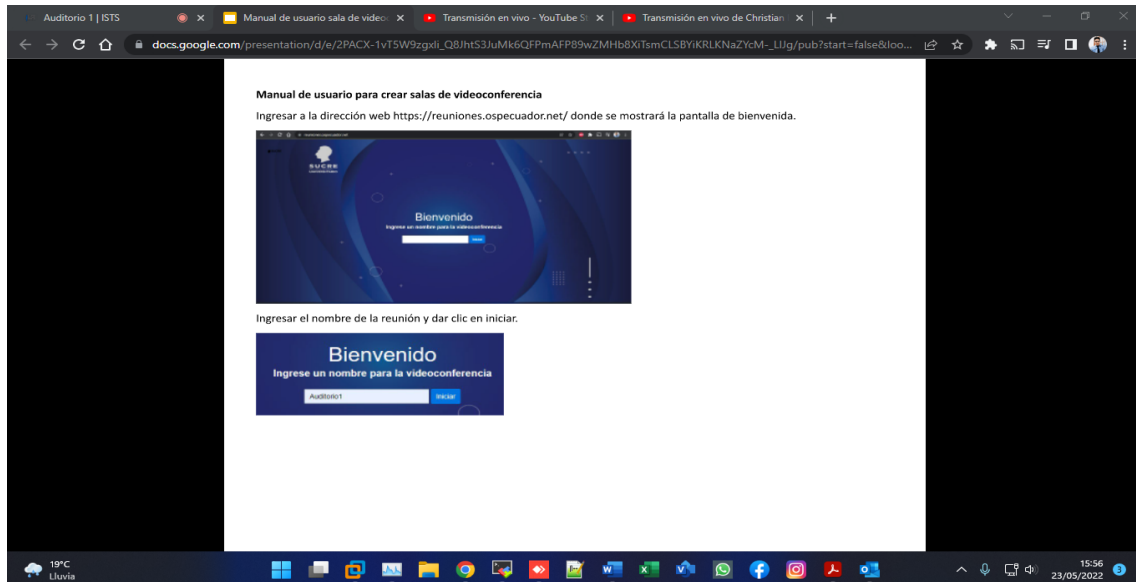


Figura 3.40. Módulo de ayuda con la explicación del funcionamiento del sistema.

3.9 PRUEBAS EN EQUIPOS MÓVILES

Al ser una aplicación de tipo web, se ha diseñado de manera *responsive*, se adapta a la resolución del equipo en el que se esté ejecutando el proceso, por lo tanto, no presentó dificultades de adaptabilidad a los equipos con los que se realizó la prueba, smartphone con Android 12, figura 3.20.

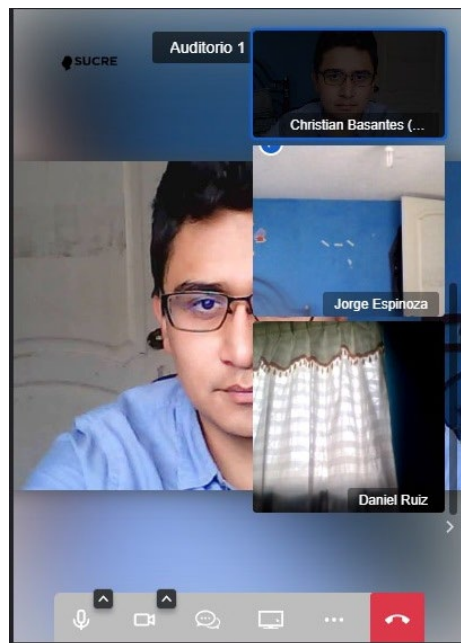


Figura 3.41. prueba realizada en smartphone con Android OS 12.

4. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- El prototipo desarrollado permite la comunicación mediante videoconferencia y la compartición de pantalla y archivos mediante el uso de API de WebRTC. Adicional, el sistema cuenta con un soporte de base de datos para el almacenamiento, asignación de roles en los usuarios y habilitación de componentes adicionales.
- El uso de la metodología KANBAN fue de gran ayuda al facilitar el desarrollo del prototipo debido a la estructura gráfica que permitió centrarse en las actividades y tareas en cada uno de los procesos sin dejar espacios vacíos entre ellos.
- A través del uso de diagramas de actividad y secuencia de datos, se logró diseñar de manera efectiva los módulos del sistema y su interconexión mediante la implementación de diagramas UML. Esto ha permitido que el sistema funcione de manera adecuada y que los módulos trabajen de manera interrelacionada.
- Los resultados de las pruebas demostraron que el prototipo de videoconferencia desarrollado cumple de manera satisfactoria con su función, ya que todos los módulos programados se ejecutaron al 100%. Además, el prototipo ofrece todas las opciones que se encuentran en servicios de pago, lo que significa que se ha cumplido con el objetivo planteado al inicio del proyecto de titulación.
- La tecnología WebRTC es capaz de proporcionar comunicación en tiempo real entre diferentes usuarios utilizando una arquitectura de cliente-servidor, tanto en parejas como en grupos de varios usuarios. Esto hace que sea una opción muy útil para aplicaciones de comunicación en línea.
- La configuración de los servidores se debe realizar en un entorno real para evitar el colapso del sistema, cabe mencionar que para las pruebas los servidores virtualizados funcionaron de manera óptima por no tener gran tráfico de datos.

- El rendimiento de una videoconferencia en cuanto a la cantidad de usuarios que pueden unirse depende directamente del ancho de banda contratado, en la red local se limita al ancho de banda de la red interna. Sin embargo, es importante tener en cuenta que la cantidad de usuarios que pueden unirse no está limitada por el prototipo en sí, sino por el ancho de banda disponible.
- Utilizar la tecnología WebRTC en navegadores web para dispositivos con sistemas operativos Android y iOS puede ser una excelente opción en lugar de crear y publicar aplicaciones específicas en las tiendas de aplicaciones de estos sistemas. Esto puede simplificar el proceso de desarrollo y distribución de aplicaciones de comunicación en línea y hacerlas más accesibles para los usuarios.
- Los resultados obtenidos de la prueba realizada en el marco del trabajo de titulación indican que el sistema de videoconferencia implementado presenta un bajo consumo de memoria RAM y procesador, lo cual es un aspecto positivo en términos de eficiencia y rendimiento del sistema. No obstante, se pudo observar que el ancho de banda utilizado durante la prueba fue la mitad del ancho de banda contratado en el instituto, lo que puede ser una limitación en términos de costos y capacidad de conexión.

4.2 RECOMENDACIONES

- Se recomienda tener en cuenta la compatibilidad de los módulos de Node.js con el sistema operativo en el que se está ejecutando. Algunos sistemas operativos, como Linux, pueden presentar incompatibilidades con ciertos módulos de Node.js, especialmente en lo que respecta a los mantenimientos y actualizaciones, se debe verificar la compatibilidad de los módulos antes de utilizarlos para asegurar que el sistema funcione de manera óptima y sin problemas.
- Se recomienda instalar un servidor DNS para redirigir el nombre de dominio hacia la IP local del servidor de videoconferencia en caso de que el aplicativo web sea utilizado dentro de la red local. De esta manera, se evita el uso del ancho de banda de internet y se aprovecha al máximo el rendimiento de la red local. Esto puede ser especialmente útil en entornos con un ancho de

banda limitado o para mejorar la velocidad y la eficiencia de la videoconferencia.

- Se recomienda utilizar una red cableada para mejorar el rendimiento de la videoconferencia y evitar problemas como latencia al compartir pantalla, desconexión y baja calidad de video. El uso de una red cableada puede ayudar a garantizar una conexión más estable y de mayor velocidad, lo que se traduce en una experiencia de videoconferencia más fluida y sin problemas.
- Se recomienda mantener actualizado el motor WebRTC instalado en el servidor de videoconferencias, ya que la tecnología WebRTC en los navegadores web suele experimentar cambios regularmente. Si se producen cambios significativos en la tecnología y el motor WebRTC instalado en el servidor no está actualizado, es posible que el prototipo de videoconferencia deje de funcionar correctamente.
- Se recomienda hacer un análisis de costo beneficio si desea mover el servidor instalado a un servidor en la nube, ya que los servicios en la nube ofrecen un mayor ancho de banda en comparación al servicio de internet empresarial que suele ser muy costoso y también se debería considerar el costo del servidor virtual.
- Se recomienda mantener actualizados los navegadores web tanto en computadoras como en dispositivos móviles para aprovechar al máximo la tecnología WebRTC. Esto se debe a que WebRTC experimenta cambios constantes que pueden mejorar la experiencia de uso de esta tecnología. Al utilizar un navegador web actualizado, es más probable que se obtenga un rendimiento óptimo y se aprovechen todas las características y mejoras disponibles.
- Se recomienda contar con hardware que cumpla con al menos 8 núcleos en el procesador, 16 GB de RAM y 50 GB de disco duro. Para garantizar un rendimiento óptimo al utilizar la tecnología WebRTC para videoconferencias a gran escala. Esto ayudará a asegurar que el sistema tenga la capacidad de manejar el tráfico de datos y el procesamiento necesario para soportar

una videoconferencia con muchos usuarios sin problemas. Es importante tener en cuenta que, a medida que aumenta el tamaño de la videoconferencia, también puede aumentar la carga en el sistema, por lo que es recomendable contar con hardware de alta calidad y de buen rendimiento para garantizar una experiencia fluida y sin problemas.

- Se recomienda realizar respaldos periódicos de las configuraciones y la base de datos del sistema de videoconferencia para minimizar el riesgo de pérdida de información en caso de que sea necesario hacer modificaciones o actualizaciones. Al hacer respaldos regularmente, se puede garantizar que se tenga acceso a una copia de seguridad de la información del sistema en caso de que se produzca algún problema. De esta manera, es posible minimizar el tiempo y el esfuerzo necesarios para recuperar la información y volver a poner en funcionamiento el sistema de videoconferencia.
- Se recomienda mantener habilitados únicamente los puertos necesarios y cerrar todos los demás para proteger el servidor y garantizar una mayor seguridad.
- Se recomienda para mejorar la disponibilidad, confiabilidad y confidencialidad se recomienda realizar una evaluación periódica del hardware y software del servidor de videoconferencia para asegurar su correcto funcionamiento y prevenir posibles fallas técnicas. Implementar un sistema de respaldo de energía eléctrica para el servidor que permita mantener su operatividad en caso de cortes de energía eléctrica o fallas en el suministro eléctrico. Configurar un sistema de monitoreo remoto del servidor de videoconferencia para identificar posibles problemas o fallas en tiempo real, y realizar las acciones correctivas necesarias de forma oportuna. Establecer medidas de seguridad para garantizar la confidencialidad de las comunicaciones realizadas a través del servidor de videoconferencia, como el uso de protocolos de encriptación y autenticación de usuarios. Desarrollar un plan de contingencia para enfrentar situaciones imprevistas, tales como ataques informáticos, fallas en los sistemas de seguridad, entre otros. Realizar capacitaciones periódicas a los usuarios del servidor de videoconferencia para fomentar buenas prácticas en cuanto a la seguridad

de la información, el uso adecuado de la plataforma y la identificación de posibles situaciones de riesgo. Mantener una comunicación constante con los proveedores de servicios de soporte técnico y de seguridad informática para mantener el servidor de videoconferencia actualizado y adaptado a las últimas tendencias en tecnología y seguridad.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] C. Mendoza, *Diseño e implementación de un prototipo Web para comunicación en tiempo real que permite realizar video llamadas, transferencia de archivos y chat, mediante el uso de WebRTC.*, Quito: Escuela Politécnica Nacional., 2016.
- [2] C. Echeverre, *DESarrollo de un aplicación Web para mejorar el control de avances académicos y asistencia de docentes de la UNT, Sub Sede Valle Jequetepeque*, Trujillo: Universidad Nacional de Trujillo, 2021.
- [3] E. Iglesias, *GOWEXSwap, Gestor de publicidad*, Madrid: Universidad Politécnica de Madrid, 2018.
- [4] F. Martín, *Desarrollo de un sistema de comunicaciones basado en WebRTC.*, Leganés: Universidad Carlos III de Madrid, 2016.
- [5] J. Delgado, *Estudio de WebRTC y su implementación con J2EE/JavaEE*, valladolid: Universidad de Valladolid, 2019.
- [6] M. Peralbo, *DESARROLLO DE UNA APLICACIÓN WEB ALTERNATIVA PARA VIDEOCONFERENCIA Y COMPARTICIÓN DE PANTALLA CON EL USO DE WEBRTC*, Quito, 2019.
- [7] B. Jasen, «Performance Analysis of WebRTC-based Video Conferencing,» *Network Architectures and Services Group*, vol. 86, pp. 1-90, 2016.
- [8] Z. Danyll, *ESTUDIO DE LAS CARACTERISTICAS DE NUEVAS ARQUITECTURAS WEB BASADAS EN WEBRTC ALOJADA EN LA NUBE Y FACTIBLE IMPLEMENTACION PARA APLICACIONES DE VOZ SOBRE IP (VoIP)*, Quito, 2015.
- [9] J. V. R. Delgado, «uva.doc,» 2016. [En línea]. Available: <https://uvadoc.uva.es/bitstream/handle/10324/18251/TFG-B%20905.PDF;jsessionid=24F73333BFD356765EDCEF0CF0F6980F?sequence=1>. [Último acceso: 11-08-2022]

- [10] J. Ortiz, «Vp8 Códec de Video,» 1 Marzo 2018. [En línea]. Available:
] <https://javierortiz.mx/glosario/códec-de-video/códec-video-vp8/>.
- [11] R. Bloor, «¿Qué es g.711? - definición de techopedia - Redes - 2022,» 1 1 2022. [En línea]. Available: <https://es.theastrologypage.com/g-711>.
- [12] U. Ipi, «Códecs,» [En línea]. Available:
https://www.lpi.tel.uva.es/~nacho/docencia/ing_ond_1/trabajos_07_08/io5/public_html/códecs_voz/ilbc.htm.
- [13] F. Michel, «WebRTC Video Conferencing, In What Browser? Take VI,» 1 3 2020. [En línea]. Available: <https://www.nojitter.com/webrtc-video-conferencing-what-browser-take-vi>.
- [14] S. Peter, S. Kevin y T. Remko, XMPP: The Definitive Guide, Tokio: Boykma, 2019.
- [15] Open JS FUndation, «Node Js,» 1 1 2020. [En línea]. Available: <https://nodejs.org/es/>.
- [16] G. Lim, Beginning Node.js, Express & MongoDB Development, Londres: Greg Lim, 2019.
- [17] Cristian Jonathan, Ronda Sangovalin , «Repositorio EPN,» Enero 2020. [En línea]. Available: <file:///C:/Users/annet/Downloads/CD%2010200.pdf>.
- [18] B. Eisenman, oreilly learning react active, United States : O´Reilly media , 2015.
- [19] A. Libby, Instant SAAS CSS how-to, Birmingham: Packt Publishing Ltd, 2013.
- [20] Leankor, Kanban for dummies, France: Leankor, 2017.
- [21] L. Castellano, «Kanban. Metodología para aumentar la eficiencia de los procesos.,» *3C Tecnología.*, vol. 8, nº 1, pp. 30-40, 2019.
- [22] C. Roca, «The Power Business School,» 18 febrero 2018. [En línea]. Available:
<https://www.thepowermba.com/es/blog/metodologia-kanban>. [Último acceso: 11 07 2022].

- [23] M. J. Fernández, J. M. Angós y J. Salvador, «Interfaces de usuario: Diseño de la visualización de la información como medio para mejorar la gestión del conocimiento y los resultados obtenidos por el usuario.,» V congreso ISKO, Madrid, 2016.
- [24] Mendoza Velásquez, Hernán Fabricio , «Repositorio Espe,» 25 Enero 2021. [En línea]. Available: <http://repositorio.espe.edu.ec/jspui/bitstream/21000/23416/1/T-ESPE-044197.pdf>.
- [25] GitHub, «¿Por qué Git Hub?,» 2021. [En línea]. Available: https://github-com.translate.google/socketio/socket.io?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=en-US&_x_tr_pto=nui,sc.
- [26] J. Delgado, Estudio de WebRTC y su implementación con J2EE/JavaEE, Valladolid, 2016.
- [27] Strap, «¿Qué es una Aplicación Web?,» [En línea]. Available: <https://www.strappinc.com/blog/strapp-datos/que-es-una-aplicacion-web>.

ANEXOS

ANEXO A

1 ENCUESTA REALIZADA PARA DETERMINAR LOS REQUERIMIENTOS DEL SISTEMA DE VIDEOCONFERENCIA.

Cabe mencionar que la encuesta realizada a los docentes del ITS se realizó de manera virtual, a través de formulario de Google Forms. A continuación, se presenta el capture de pantalla con el formulario enviado, a los docentes de la carrera de diseño de software, y su respectiva tabulación.

ESCUELA POLITÉCNICA NACIONAL

Requerimientos de sistema.

Esta encuesta tiene como finalidad la obtención de datos para los requerimientos del sistema de Videoconferencia que se desarrollará para el Instituto Tecnológico Sucre, encuesta dedicada al cuerpo docente.

¿Con que frecuencia usted realiza videoconferencias? *

Poca Frecuencia 1 2 3 4 5 Mayor Frecuencia

¿La aplicación actual cumple con los requerimiento que usted necesita? *

Sí
 No

¿Qué considera usted que debería tener una aplicación propia ? *

Transmisión en alguna plataforma externa
 Compartición de pantalla
 Admisión de participantes
 Control sobre las opciones disponible

¿Considera que el pago de una licencia anual debería suspenderse para los institutos académicos? *

Sí
 No

¿Cuánta acogida cree que podría tener una aplicación propia para el ITS? *

Poca acogida 1 2 3 4 5 gran Acogida

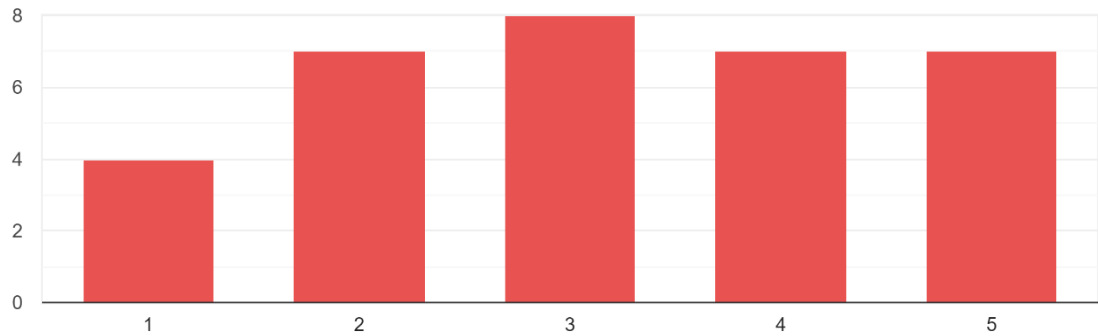
¿Cambiaría el uso de la plataforma actual por una propia con sus estudiantes del ITS? *

Sí
 No
 Tal vez

Enviar Borrar formulario

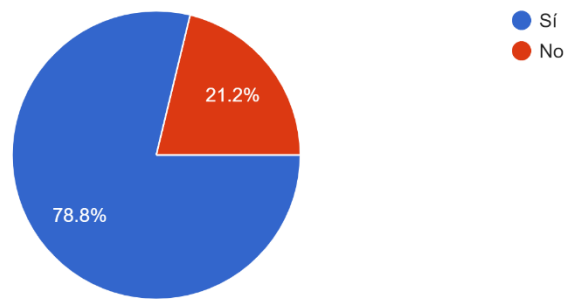
¿Con que frecuencia usted realiza videoconferencias?

33 respuestas



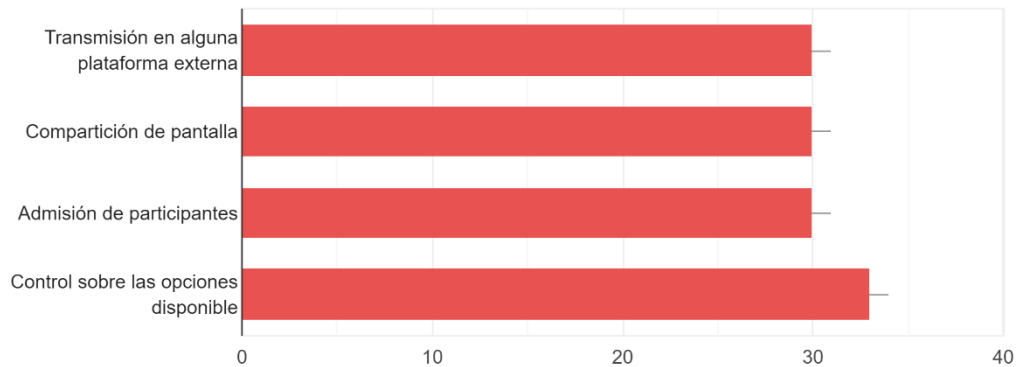
¿La aplicación actual cumple con los requerimiento que usted necesita?

33 respuestas

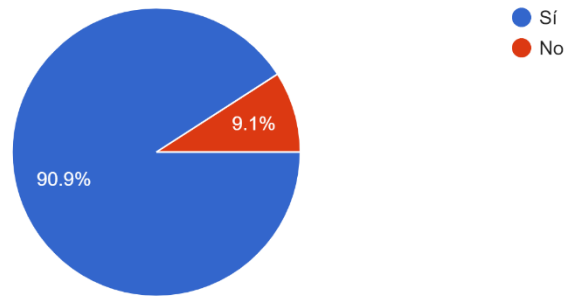


¿Qué considera usted que debería tener una aplicación propia ?

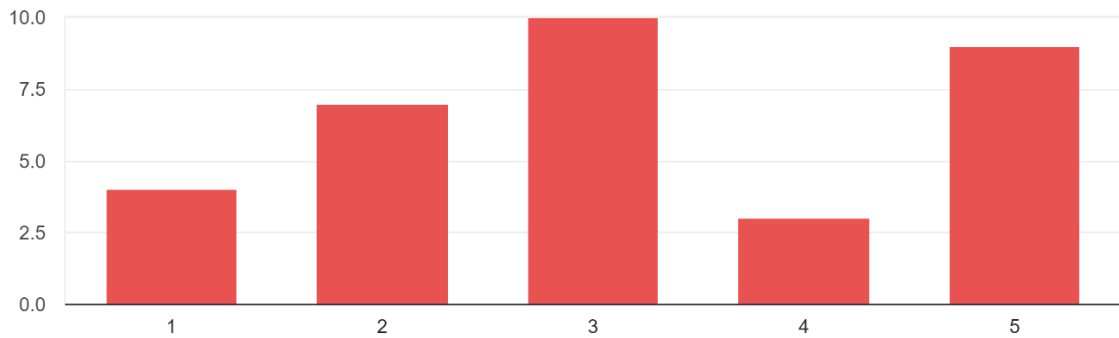
33 respuestas



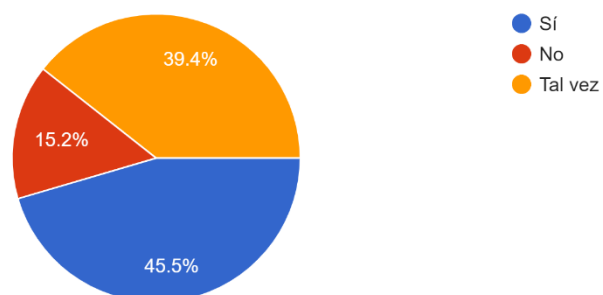
¿Considera que el pago de una licencia anual debería suspenderse para los institutos académicos?
33 respuestas



¿Cuánta acogida cree que podría tener una aplicación propia para el ITS?
33 respuestas




¿Cambiaría el uso de la plataforma actual por una propia con sus estudiantes del ITS?
33 respuestas



2 ENCUESTA REALIZADA PARA DETERMINAR LA SATISFACCIÓN DEL USO DEL PROTOTIPO.

Al igual que la encuesta previa para determinar los requisitos, esta también se realizó a los docentes del ITS que utilizaron el sistema de videoconferencia, a través de formulario de Google Forms. A continuación, se presenta el capture de pantalla con el formulario enviado, a los docentes de la carrera de diseño de software, y su respectiva tabulación.

 ESCUELA POLITÉCNICA NACIONAL

Encuesta de satisfacción.

Esta encuesta tiene como finalidad la obtención de información referente al cumplimiento y satisfacción del prototipo Videoconferencia desarrollado para el Instituto Tecnológico Suora, encuesta dedicada al cuerpo docente.

¿Con que frecuencia usted usaría el prototipo videoconferencia desarrollado? *

Poca Frecuencia 1 2 3 4 5 Mayor Frecuencia

¿La aplicación actual cumple con los requerimiento que usted necesita? *

Si

No

¿Existió problemas para el ingreso al prototipo de videoconferencias? *

Si

No

¿Cree que el acceso dentro del prototipo es fácil de entender? *

Si

No

¿Cuánta acogida cree que tendría el prototipo diseñado para el ITS? *

Poca acogida 1 2 3 4 5 gran Acogida

¿Cambiaría el uso de la plataforma actual por este prototipo con sus estudiantes del ITS? *

Si

No

Tal vez

¿Existió algún problema con alguna de las aplicaciones internas? *

Compartir pantalla

Audio

Micrófono

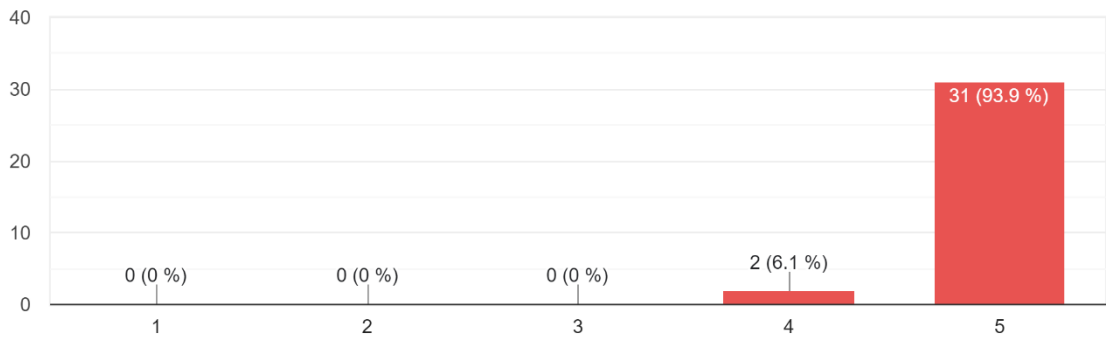
Cámara

Transmisión en plataforma YouTube

Ninguno

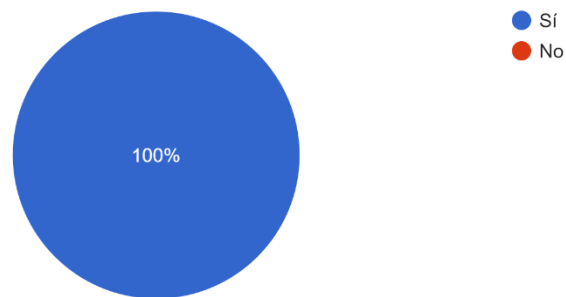
¿Con que frecuencia usted usaría el prototipo videoconferencia desarrollado?

33 respuestas



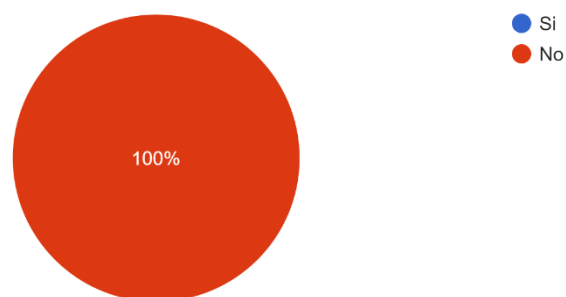
¿La aplicación actual cumple con los requerimiento que usted necesita?

33 respuestas



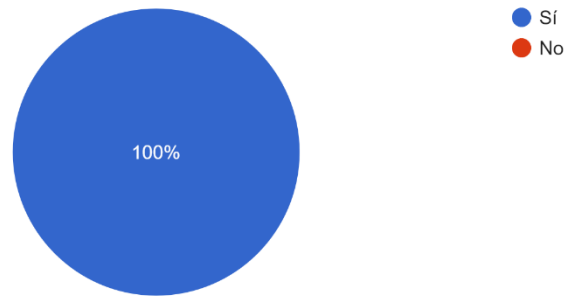
¿Existió problemas para el ingreso al prototipo de videoconferencias?

33 respuestas



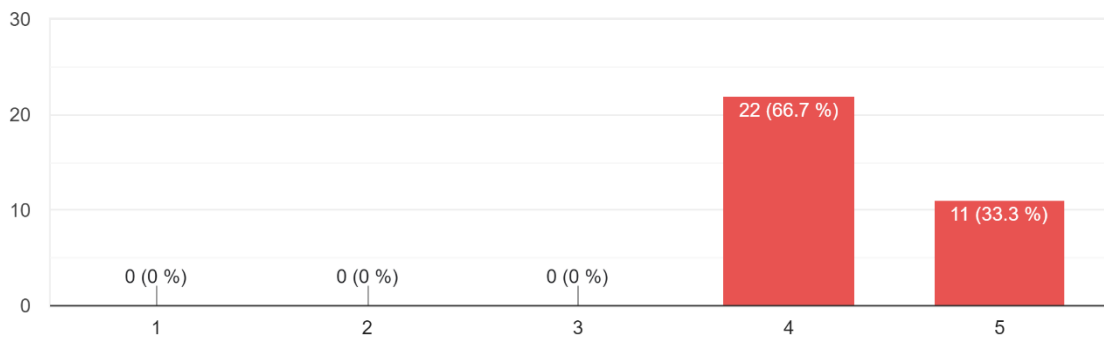
¿Cree que el acceso dentro del prototipo es fácil de entender?

33 respuestas



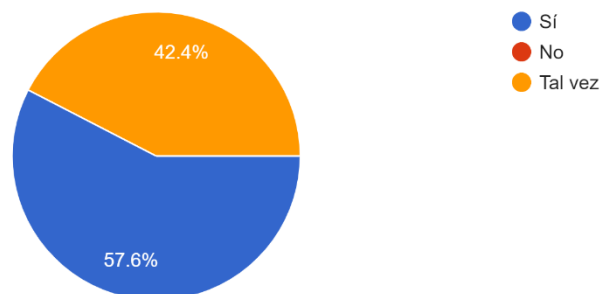
¿Cuánta acogida cree que tendría el prototipo diseñado para el ITS?

33 respuestas



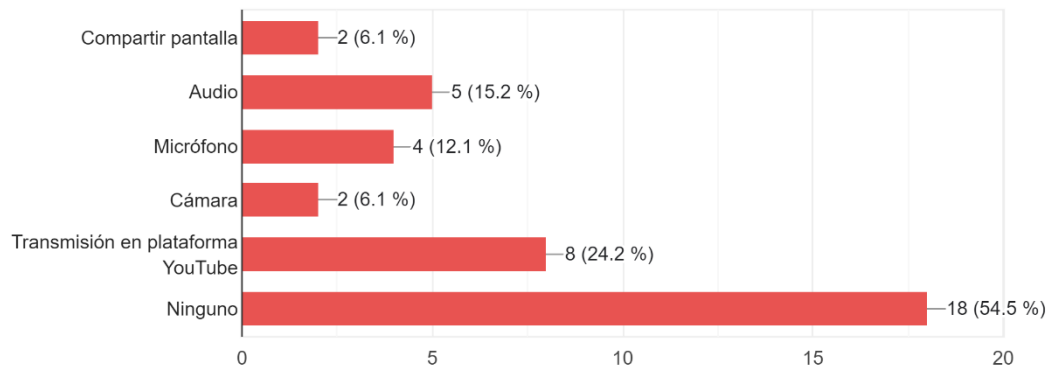
¿Cambiaría el uso de la plataforma actual por este prototipo con sus estudiantes del ITS?

33 respuestas



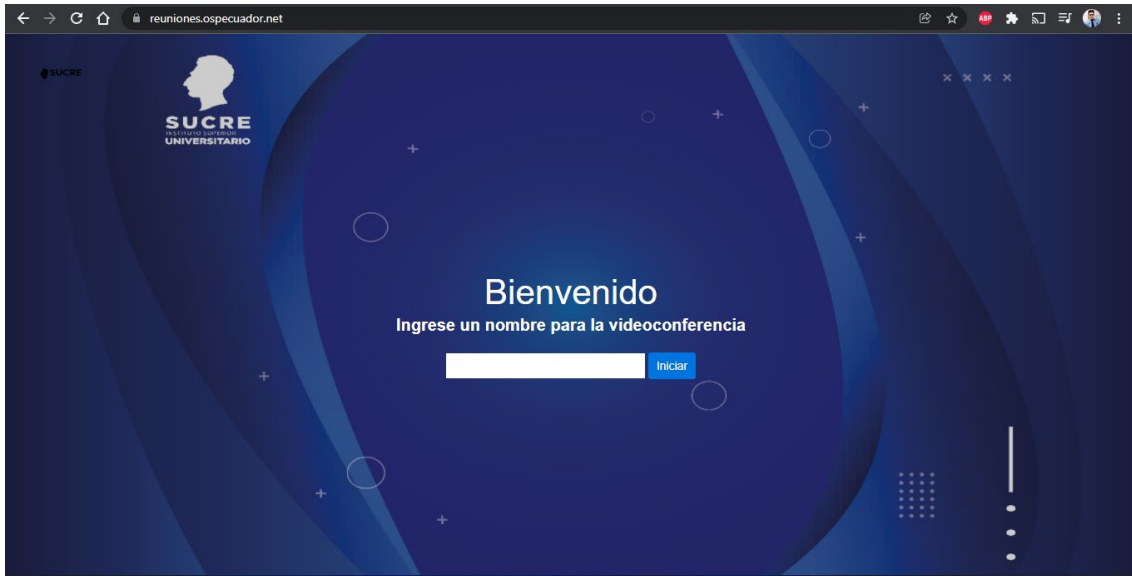
¿Existió algún problema con alguna de las aplicaciones internas?

33 respuestas

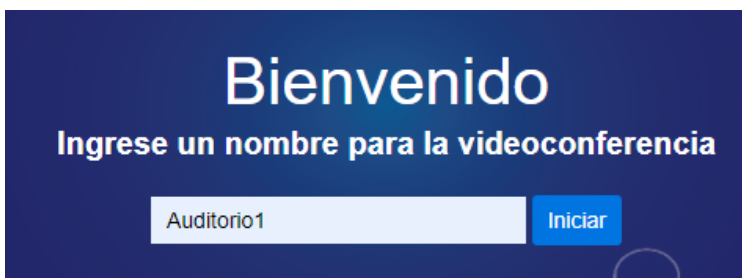


3 MANUAL DE USUARIO PARA CREAR SALAS DE VIDEOCONFERENCIA

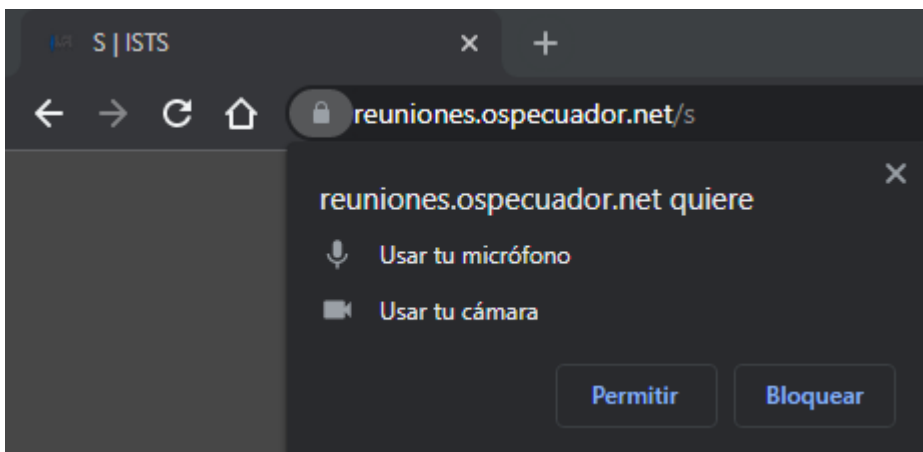
Ingresar a la dirección web <https://reuniones.ospecuador.net/> donde se mostrará la pantalla de bienvenida.



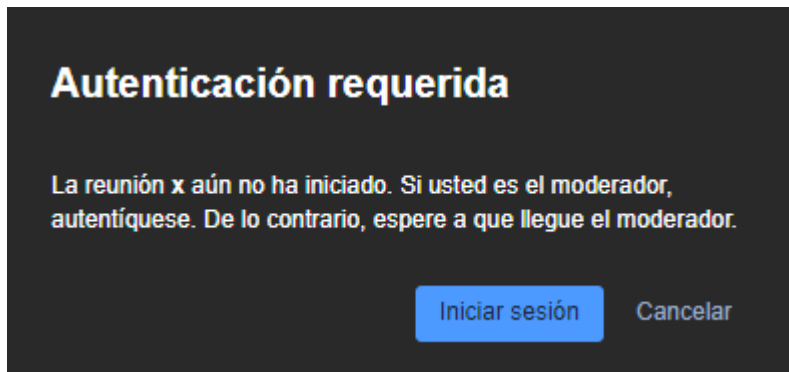
Ingresar el nombre de la reunión y dar clic en iniciar.



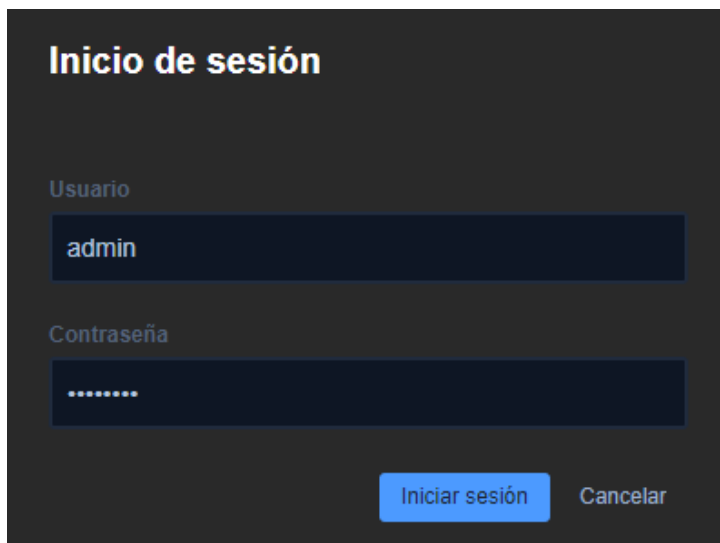
Al acceder a la plataforma por primera vez nos preguntará si queremos conceder permisos para acceder al micrófono y la cámara web, damos clic en permitir.



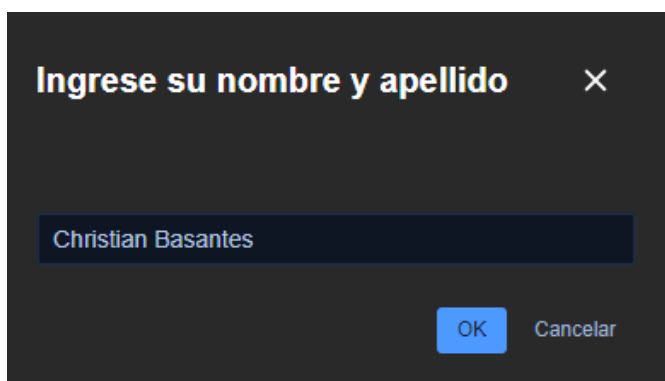
Nos saldrá un mensaje de espera al moderador de la reunión ya que aún la sala de videoconferencia aún no se ha creado, por lo que es necesario iniciar sesión.



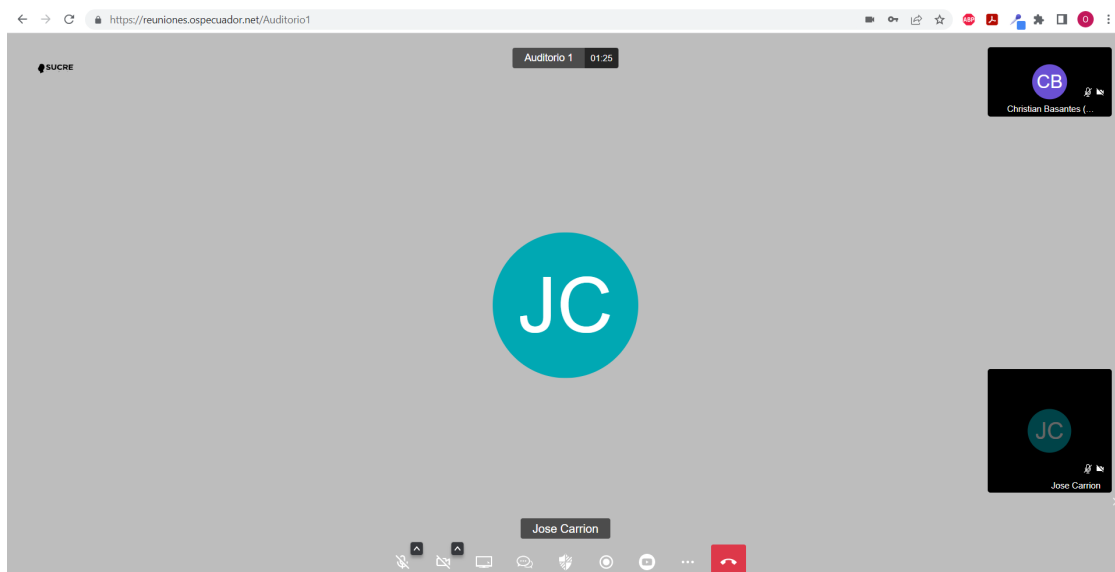
Nos pedirá nombre de usuario y contraseña luego clic en iniciar sesión.



Nos pedirá nombre y apellido como nombre de usuario para mostrar a todos los participantes.



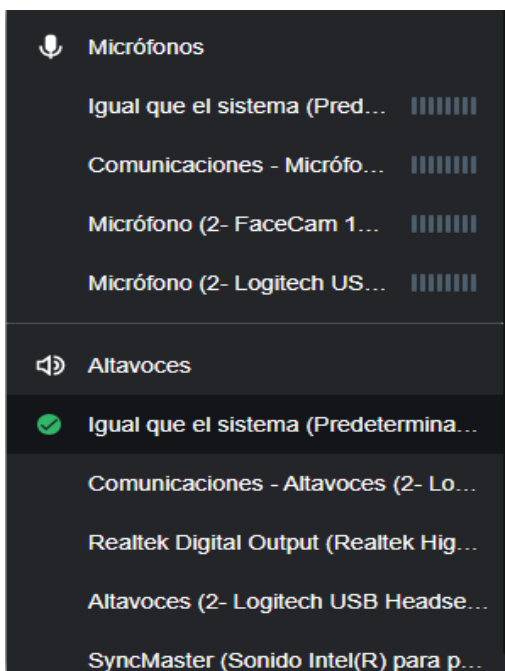
Se habrá creado la sala de videoconferencia, por defecto la cámara y micrófono estarán apagadas para evitar interrupciones cuando ingrese más participantes.



En caso de tener más dispositivos como micrófonos y altavoces, damos clic en configuración de audio.



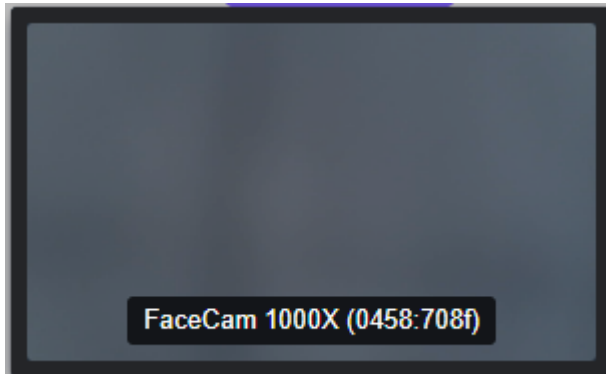
Aquí escogeremos con un clic el micrófono y altavoz a utilizar.



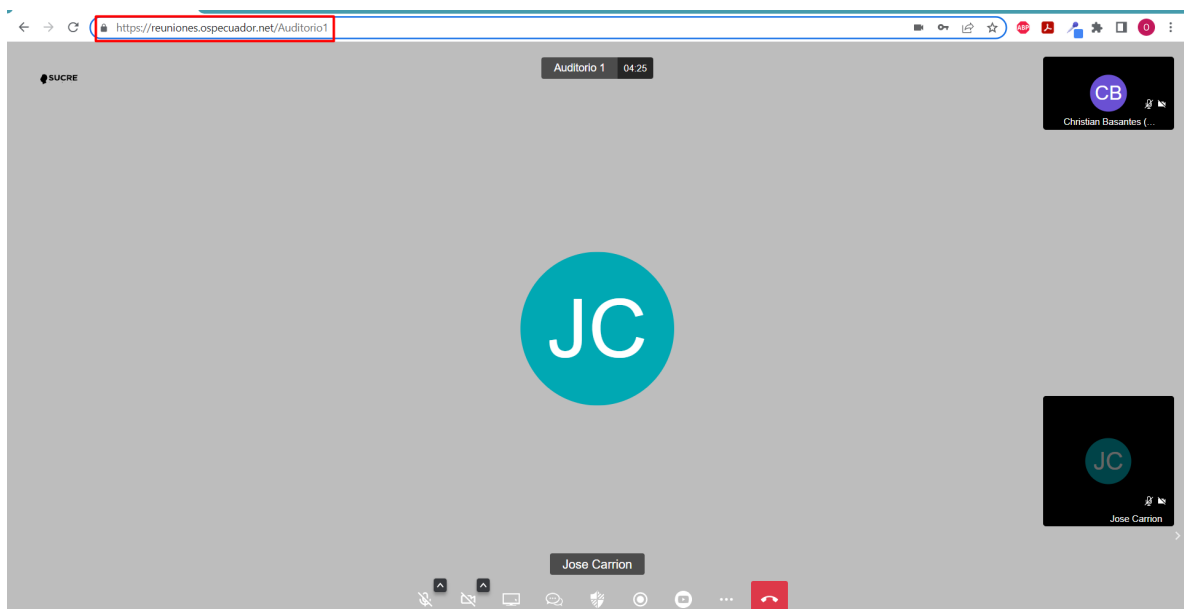
En caso de tener más dispositivos como cámaras web, damos clic en configuración de video.



Aquí escogeremos con un clic la cámara web a utilizar.



Para invitar a más participantes a la videoconferencia es necesario compartir el enlace de la videoconferencia, donde aparecerán los participantes con su nombre en una barra lateral derecha.



En la barra de herramientas tenemos los siguientes botones:

Activar o desactivar el micrófono.



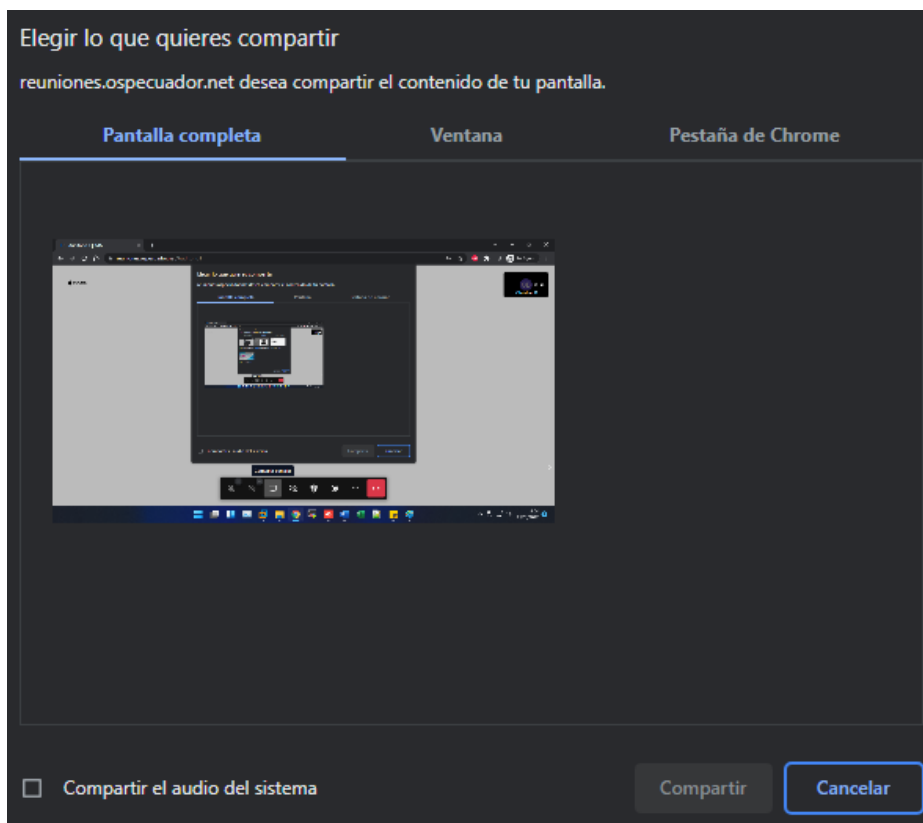
Activar o desactivar la cámara web.



Compartir pantalla a todos los participantes.



En compartir pantalla podremos compartir la pantalla completa, una ventana específica o una pestaña del navegador; también se puede compartir o no el audio de la pantalla seleccionada.



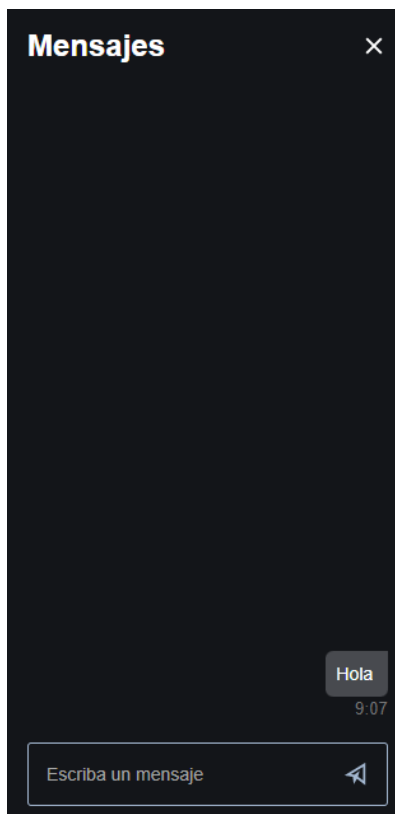
Para dejar de compartir la pantalla hacemos clic de nuevo en compartir pantalla.



Envío de mensajes a todos los participantes.



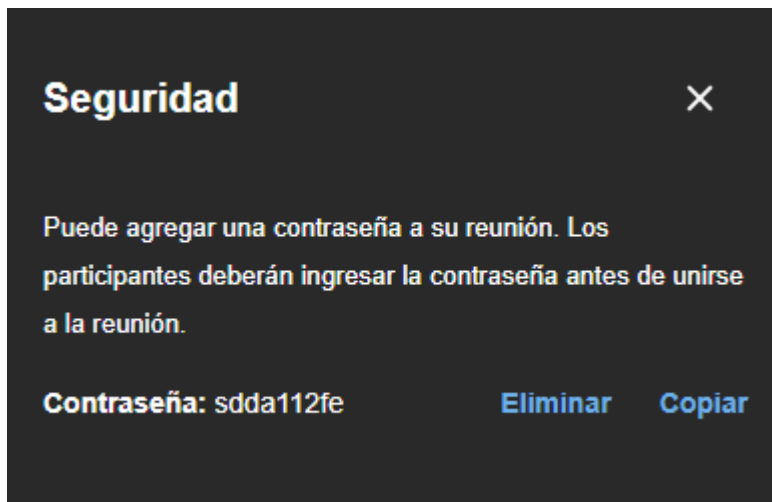
Se desplegará una barra lateral izquierda de mensajes donde podrán interactuar todos los participantes mediante mensajes de texto.



Asignar contraseña a la sala de videoconferencia.



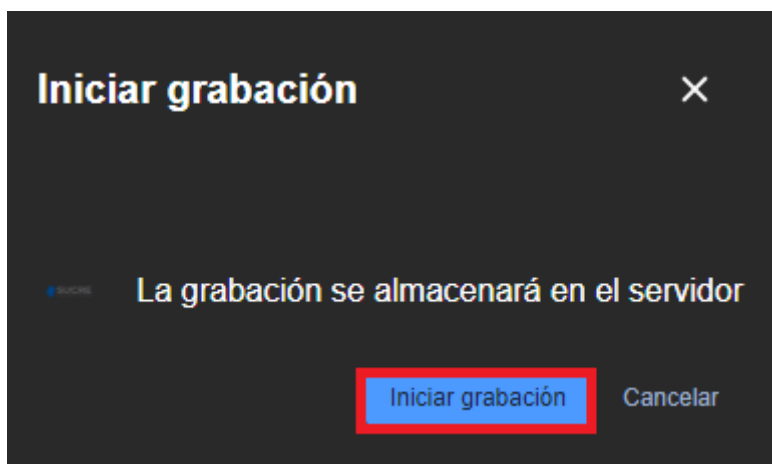
Aparecerá la siguiente pantalla para poder asignar una contraseña.



Grabar la videoconferencia en el servidor.



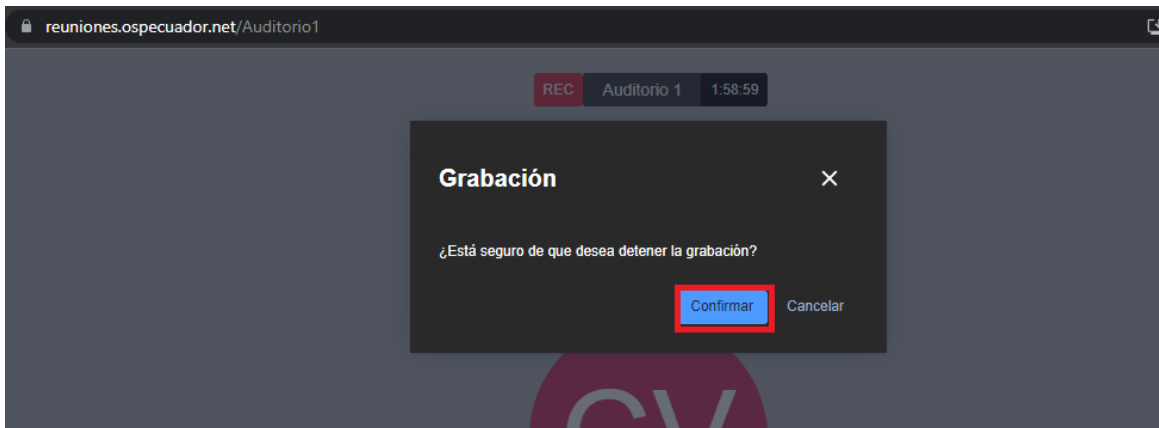
Aparecerá la siguiente pantalla para iniciar la grabación.



Para detener la grabación hacemos clic de nuevo en el icono de grabar.



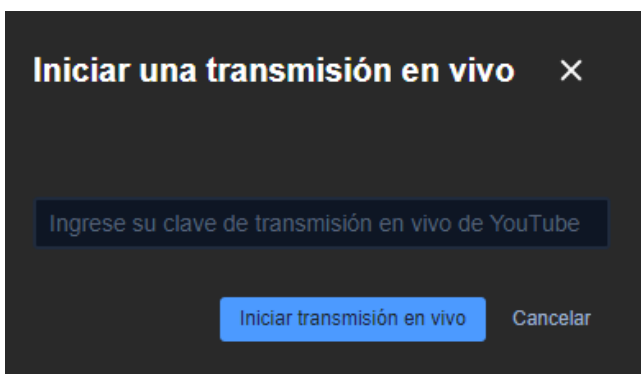
Aparecerá la siguiente pantalla donde nos pedirá la confirmación para detener la grabación.



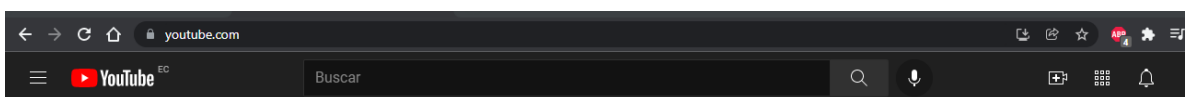
Transmitir la videoconferencia en YouTube.



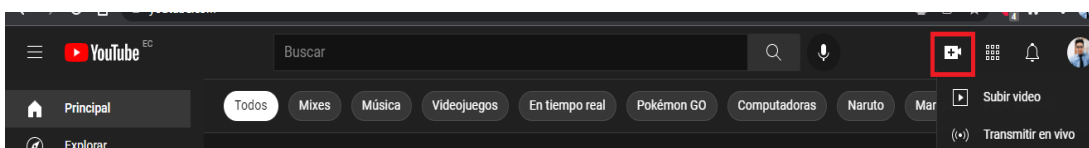
Aparecerá la siguiente pantalla solicitando la clave de transmisión de YouTube.



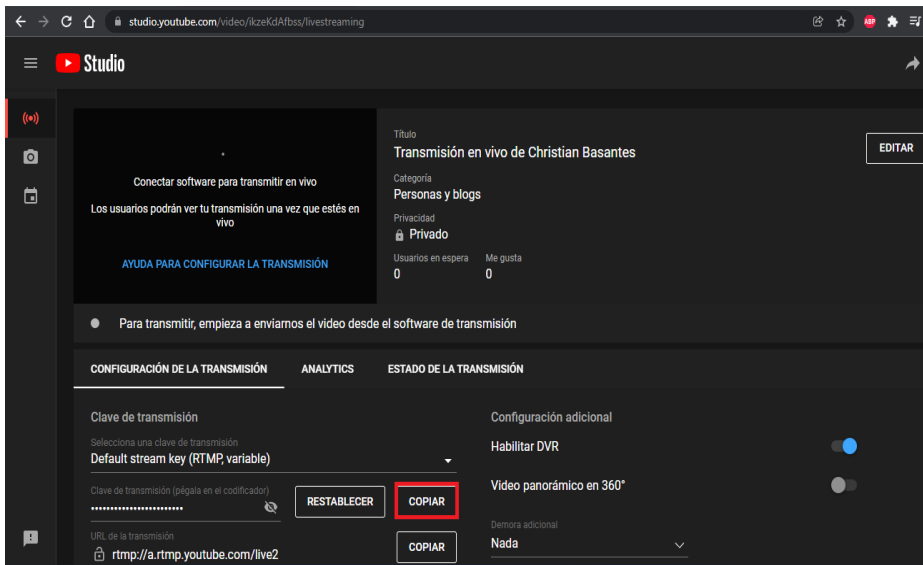
Para obtener la clave de transmisión en vivo de YouTube ingresamos a YouTube.



Damos clic en el siguiente icono luego clic en transmitir en vivo.



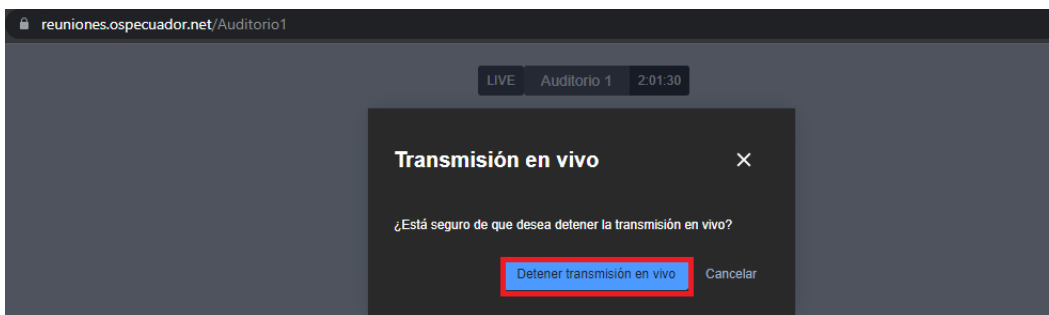
Luego clic en copiar clave de transmisión y la pegamos en la plataforma de videoconferencias.



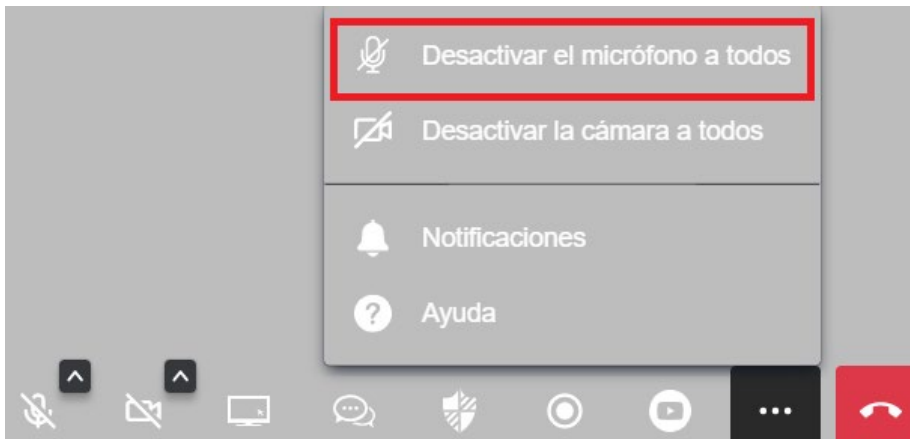
Para detener la transmisión en vivo hacemos clic en el icono de transmisión en vivo.



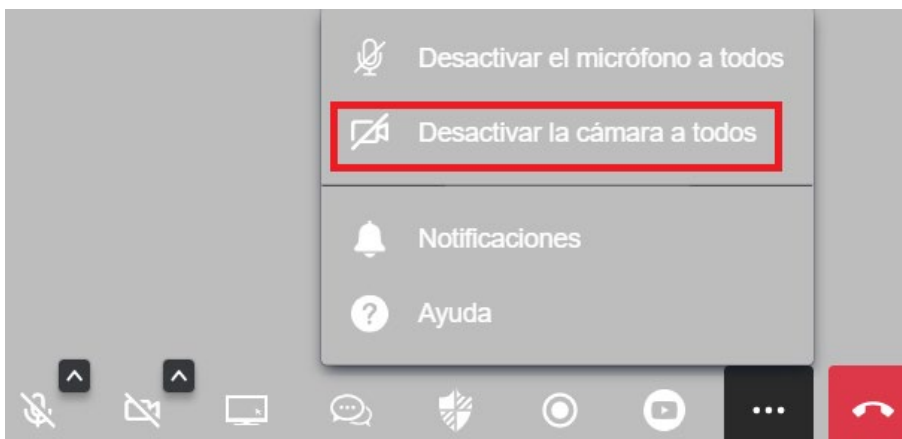
Aparecerá la siguiente pantalla de confirmación para detener la transmisión en vivo.



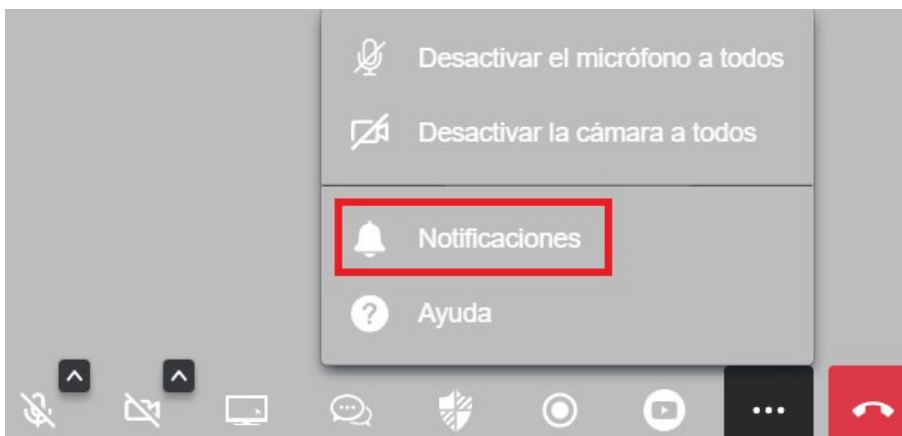
En más acciones tenemos, desactivar el micrófono a todos los participantes.



Desactivar la cámara a todos los participantes.

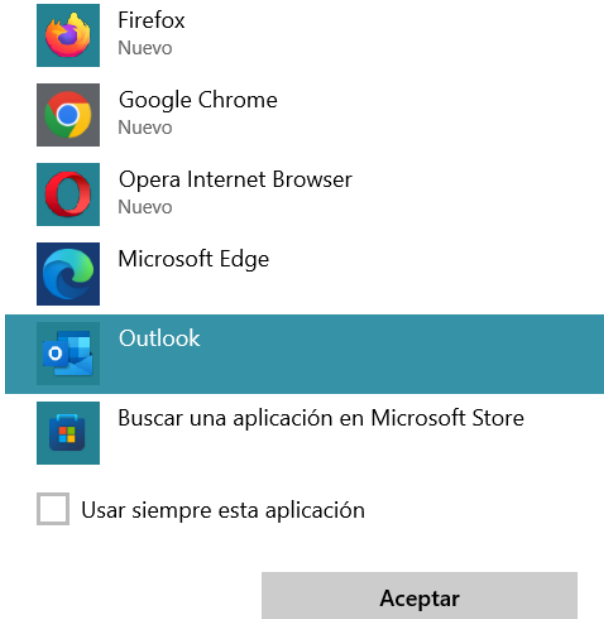


Enviar notificaciones de inconvenientes con la plataforma de videoconferencia.

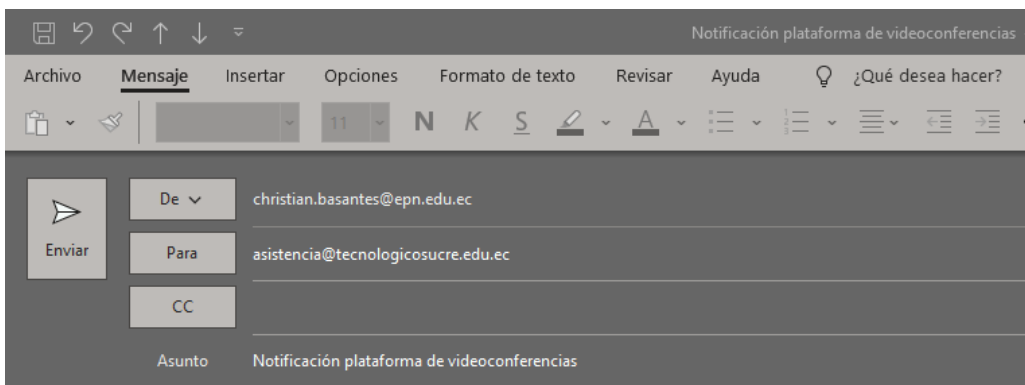


Se desplegará una pantalla para escoger nuestro gestor de correo.

¿Cómo quieres abrir esto?



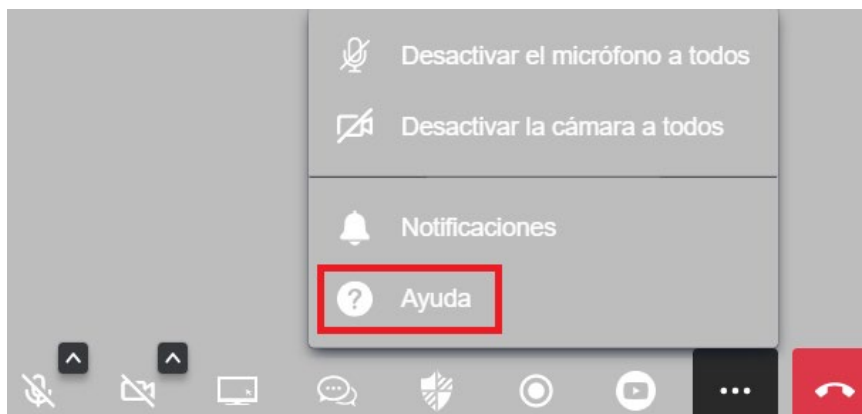
Se abrirá el gestor de correo con el campo de destinatario y asunto de forma predeterminada.



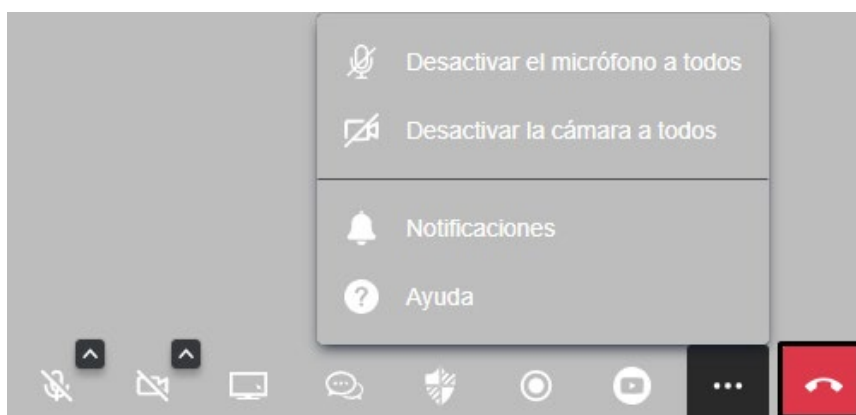
Estimados,

Saludos cordiales,

Ayuda de funcionalidades de la plataforma de videoconferencia.



Salir de la reunión



4 MANUAL DE ADMINISTRADOR PARA LA ADMINISTRACIÓN DE MODERADORES

Ingresar a la dirección web <http://administs.ospecuador.net:8080/administs/> donde se mostrará la pantalla de bienvenida.



Clic en iniciar sesión e ingresamos las credenciales de administrador único.

[Inicio](#)

INICIO DE SESIÓN

Usuario:

Contraseña:

Nos dará la bienvenida con nuestro nombre, donde podremos cerrar sesión realizar las operaciones de CRUD créate read update delete de moderadores.

ADMINISTRACIÓN DE USUARIOS PROTOTIPO DE VIDEOCONFERENCIAS

SUCRE
INSTITUTO SUPERIOR
UNIVERSITARIO

Bienvenido, Christian Basantes [Cerrar sesión](#)

[ADMINISTRAR MODERADORES](#) | [AGREGAR ADMINISTRADOR](#)

INICIAR VIDEOCONFERENCIA DE [PRUEBA](#)

En administrar moderadores tendremos las opciones de volver al inicio, agregar moderador, cerrar sesión y buscar un moderador mediante cédula

Usuario	Contraseña	Actualizar
1725451742	paradise	Editar contraseña Eliminar
1725451749	22315551515	Editar contraseña Eliminar
1745789685	mksdasweE33	Editar contraseña Eliminar
1745789865	s22d5e8ew77	Editar contraseña Eliminar
1745874410	761ze81qt7lw	Editar contraseña Eliminar
1745986325	s544f5r8e	Editar contraseña Eliminar
1747894702	pruebareunion2021	Editar contraseña Eliminar
1748786932	calindobienlindo04	Editar contraseña Eliminar
1765234798	pedroj2021	Editar contraseña Eliminar
1769845698	EdgarJ**	Editar contraseña Eliminar
1801537448	holaredes	Editar contraseña Eliminar

En agregar moderador deberemos ingresar la cédula y contraseña del moderador y damos clic en agregar

Ingrese el usuario y contraseña

Usuario:

Contraseña:

Podemos usar la opción de búsqueda en caso de que existan muchos moderadores y se podrá actualizar la contraseña o eliminar el moderador.

Usuario	Contraseña	Actualizar
1725451742	paradise	Editar contraseña Eliminar

En editar contraseña podremos actualizar la contraseña y damos clic en actualizar

Ingrese una nueva contraseña

Usuario: 1725451742

Contraseña:

También podremos eliminar al moderador damos clic en eliminar y podremos eliminar el moderador

The screenshot shows a web browser window with the address bar displaying "administs.ospecuador.net:8080/administs/view.php". The page content includes navigation links: "Inicio", "Agregar moderador", and "Cerrar sesión". Below these is a search bar with the text "Ingrese una cédula a buscar:" and a "Buscar" button. A table lists users with columns for "Usuario" and "Contraseña". Each row contains a user ID, a password, and two links: "Editar contraseña" and "Eliminar". A modal dialog box is open in the center, displaying the text "administs.ospecuador.net:8080 dice Seguro que desea eliminar?" and two buttons: "Aceptar" and "Cancelar".

Usuario	Contraseña		
1725451742	paradise	Editar contraseña	Eliminar
1725451749	22315551515	Editar contraseña	Eliminar
1745789685	mksdaswef33	Editar contraseña	Eliminar
1745789865	s22d5e8ew77	Editar contraseña	Eliminar
1745874410	761zc81qt7lw	Editar contraseña	Eliminar
1745986325	s5d4f5r8e	Editar contraseña	Eliminar
1747894702	pruebareunion.2021	Editar contraseña	Eliminar
1748786932	calindobienlindo04	Editar contraseña	Eliminar
1765234798	pedroj2021	Editar contraseña	Eliminar
1769845698	EdgarJ**	Editar contraseña	Eliminar
1801537448	holaredes	Editar contraseña	Eliminar

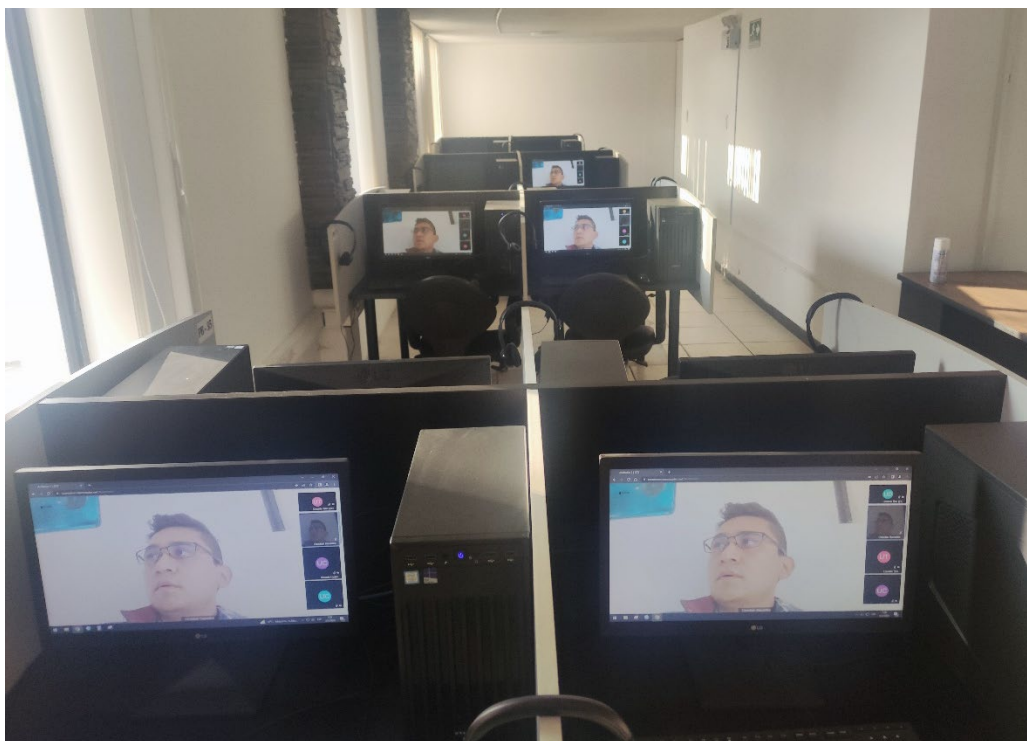
En el inicio podremos crear una sala de pruebas para probar el funcionamiento o los moderadores creados mediante el inicio de sesión de la sala de videoconferencia de prueba.



5 PRUEBAS ADICIONALES EN COMPUTADORES

Para las pruebas se creó dos salas de videoconferencias de 10 usuarios en cada sala.

Auditorio1



Auditorio2



El consumo procesador tenía un promedio de 20% de consumo y un 15% de consumo de ram.

```

top - 07:58:42 up 84 days, 15:19, 3 users, load average: 0.72, 0.38, 0.29
Tasks: 312 total, 1 running, 311 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.2 us, 1.0 sy, 0.0 ni, 97.4 id, 0.3 wa, 0.0 hi, 0.1 si, 0.0 st
MiB Mem : 15964.6 total, 5125.3 free, 2395.8 used, 8443.5 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used, 13329.8 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 1142 jvb        20   0 8998260 593764 26352 S  16.6   3.6 337:01.47 java
   40 root       20   0     0     0     0 S   0.3   0.0  0:03.85 ksoftirqd/4
 1159 influxdb  20   0 1739944 134812 27932 S   0.3   0.8 164:52.91 influxd
 1183 jicofo     20   0 9035820 583992 20024 S   0.3   3.6  97:59.15 java
 1671 prosody  20   0  39400  26960  9456 S   0.3   0.2 149:50.26 lua5.2
2321178 www-data  20   0  68920  11756  7144 S   0.3   0.1  0:08.11 nginx
2321180 www-data  20   0  67476  9912  6736 S   0.3   0.1  0:00.21 nginx
2401301 mysql     20   0 2687732 421796 40280 S   0.3   2.6 12:19.28 mysqld
2680024 root      20   0   8200   3968  3172 R   0.3   0.0  0:07.36 top
2680028 root      20   0 230080  6108  5372 S   0.3   0.0  0:01.61 iftop
   1 root      20   0 168968  13084  8332 S   0.0   0.1  8:25.92 systemd
   2 root      20   0     0     0     0 S   0.0   0.0  0:01.57 kthreadd
   3 root      0 -20   0     0     0 I   0.0   0.0  0:00.00 rcu_gp
   4 root      0 -20   0     0     0 I   0.0   0.0  0:00.00 rcu_par_gp
   5 root      0 -20   0     0     0 I   0.0   0.0  0:00.00 netns
   7 root      0 -20   0     0     0 I   0.0   0.0  0:00.00 kworker/0:0H-events_highpri
   9 root      0 -20   0     0     0 I   0.0   0.0  0:38.03 kworker/0:1H-events_highpri
  10 root      0 -20   0     0     0 I   0.0   0.0  0:00.00 mm_percpu_wq
  11 root      20   0     0     0     0 S   0.0   0.0  0:00.00 rcu_tasks_rude_
  12 root      20   0     0     0     0 S   0.0   0.0  0:00.00 rcu_tasks_trace
  13 root      20   0     0     0     0 S   0.0   0.0  0:04.45 ksoftirqd/0
  14 root      20   0     0     0     0 I   0.0   0.0 16:28.11 rcu_sched
  15 root      rt    0     0     0     0 S   0.0   0.0  0:17.75 migration/0
  16 root     -51   0     0     0     0 S   0.0   0.0  0:00.00 idle_inject/0
  18 root      20   0     0     0     0 S   0.0   0.0  0:00.00 cpuhp/0
  19 root      20   0     0     0     0 S   0.0   0.0  0:00.00 cpuhp/1
  20 root     -51   0     0     0     0 S   0.0   0.0  0:00.00 idle_inject/1
  21 root      rt    0     0     0     0 S   0.0   0.0  0:17.01 migration/1
  22 root      20   0     0     0     0 S   0.0   0.0  0:03.96 ksoftirqd/1
  24 root      0 -20   0     0     0 I   0.0   0.0  0:00.00 kworker/1:0H-events_highpri
  25 root      20   0     0     0     0 S   0.0   0.0  0:00.00 cpuhp/2

```

Consumo ancho de banda aproximado de 20 Mbps.

	191Mb	381Mb	572Mb	763Mb	954Mb	
reuniones.ospecuador.net	⇒	172.16.8.150		20.2Kb 21.7Kb 22.3Kb		
reuniones.ospecuador.net	⇒	172.16.2.155		2.97Mb 3.14Mb 3.23Mb		
reuniones.ospecuador.net	⇒	172.16.2.188		2.28Mb 2.45Mb 2.51Mb		
reuniones.ospecuador.net	⇒	172.16.2.185		11.7Kb 12.5Kb 12.9Kb		
reuniones.ospecuador.net	⇒	172.16.2.176		2.27Mb 2.44Mb 2.51Mb		
reuniones.ospecuador.net	⇒	172.16.2.173		11.7Kb 12.7Kb 13.0Kb		
reuniones.ospecuador.net	⇒	172.16.2.157		2.29Mb 2.44Mb 2.51Mb		
reuniones.ospecuador.net	⇒	172.16.2.153		12.0Kb 12.5Kb 12.9Kb		
reuniones.ospecuador.net	⇒	172.16.2.161		2.28Mb 2.44Mb 2.51Mb		
reuniones.ospecuador.net	⇒	172.16.2.177		11.1Kb 12.4Kb 12.8Kb		
reuniones.ospecuador.net	⇒	10.10.100.1		496Kb 534Kb 548Kb		
reuniones.ospecuador.net	⇒	172.16.2.156		11.0Kb 11.1Kb 11.3Kb		
reuniones.ospecuador.net	⇒	172.16.2.199		495Kb 534Kb 548Kb		
reuniones.ospecuador.net	⇒	172.16.2.158		10.2Kb 10.9Kb 11.1Kb		
reuniones.ospecuador.net	⇒	172.16.2.151		495Kb 534Kb 548Kb		
reuniones.ospecuador.net	⇒	172.16.2.194		9.61Kb 10.7Kb 11.2Kb		
reuniones.ospecuador.net	⇒			495Kb 534Kb 548Kb		
reuniones.ospecuador.net	⇒			10.2Kb 10.8Kb 11.2Kb		
reuniones.ospecuador.net	⇒			495Kb 534Kb 548Kb		
reuniones.ospecuador.net	⇒			9.70Kb 10.8Kb 11.2Kb		
reuniones.ospecuador.net	⇒			30.4Kb 40.9Kb 41.7Kb		
reuniones.ospecuador.net	⇒			31.6Kb 42.8Kb 44.1Kb		
reuniones.ospecuador.net	⇒			9.28Kb 7.63Kb 6.94Kb		
reuniones.ospecuador.net	⇒			800b 1.02Kb 1.03Kb		
reuniones.ospecuador.net	⇒			1.08Kb 1.03Kb 885b		
reuniones.ospecuador.net	⇒			1.31Kb 1.39Kb 1.23Kb		
reuniones.ospecuador.net	⇒			1.08Kb 939b 856b		
reuniones.ospecuador.net	⇒			1.53Kb 1.32Kb 1.19Kb		
reuniones.ospecuador.net	⇒			1.08Kb 883b 842b		
reuniones.ospecuador.net	⇒			1.53Kb 1.23Kb 1.21Kb		
reuniones.ospecuador.net	⇒			0b 834b 856b		
reuniones.ospecuador.net	⇒			448b 1.26Kb 1.24Kb		
Tx:	cum:	1.14GB	peak:	14.0Mb	rates:	11.6Mb 12.5Mb 12.8Mb
Rx:		359Mb		3.72Mb		3.10Mb 3.29Mb 3.39Mb
TOTAL:		1.49GB		17.7Mb		14.7Mb 15.7Mb 16.2Mb