

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA:  
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:  
Ingenieros de sistemas**

**TEMA:  
ANÁLISIS, DISEÑO, Y CONSTRUCCIÓN DE UNA APLICACIÓN WEB DE  
VIDEOCONFERENCIA UTILIZANDO EL PROTOCOLO WEB REAL-TIME  
COMMUNICATION**

**AUTORES:  
FRANKLIN STALIN QUIROGA LOZANO  
SANTIAGO ANDRÉS ROSERO IBUJÉS**

**TUTOR:  
Díaz Ortiz, Daniel Giovanny**

**Quito, febrero del 2021**

## CESIÓN DE DERECHOS DE AUTOR

Nosotros Quiroga Lozano Franklin Stalin, con documento de identificación N° 1723687628 y Rosero Ibufés Santiago Andrés, con documento de identificación N° 1724709496, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: ANÁLISIS, DISEÑO, Y CONSTRUCCIÓN DE UNA APLICACIÓN WEB DE VIDEOCONFERENCIA UTILIZANDO EL PROTOCOLO WEB REAL-TIME COMMUNICATION, mismo que ha sido desarrollado para optar por el título de: INGENIEROS DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana



.....  
QUIROGA LOZANO FRANKIN STALIN  
CI: 1723687628



.....  
ROSERO IBUJÉS SANTIAGO ANDRÉS  
CI: 1724709496

Quito, febrero del 2021

## DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el proyecto técnico, con el tema ANÁLISIS, DISEÑO, Y CONSTRUCCIÓN DE UNA APLICACIÓN WEB DE VIDEOCONFERENCIA UTILIZANDO EL PROTOCOLO WEB REAL-TIME COMMUNICATION realizado por Franklin Stalin Quiroga Lozano y Santiago Andrés Rosero Ibujés, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, febrero del 2021



.....  
Díaz Ortiz, Daniel Giovanni  
CI: 1716975501

## **DEDICATORIA**

El presente proyecto está dedicado a mi familia quien en toda esta etapa de mi vida siempre tuvo ahí con su apoyo y palabras de aliento, gracias a mi padre Franklin que del aprendí que para llegar a tener éxito en la vida uno debe ser constante en su trabajo dar lo mejor de uno y siempre ayudar a los demás cuando lo necesiten a mi madre Sonia que a pesar todo los dolores de cabeza que le ocasione siempre confió en mí y nunca dudo de los éxitos que estoy cosechando lo valores que tengo como persona se los debo a ella mi hermana Cristina que me enseñó que a pesar de tener varios obstáculos en la vida uno puede salir adelante y ser un ejemplo a seguir, mi hermana Katherin la única que conoce de todas las locuras que he tenido a ella le debo mucho por sus palabras de aliento por siempre estar ahí y robarme muchas risas en los peores días que pude tener, sus consejos lo llevo en mí corazón y gracias a eso cada día creo ser mejor persona y por último pero no la menos importante Angela que sin ella tal vez muchas veces me habría quebrado solo, ella me ayudo en toda la carrera a seguir adelante a no darme por vencido y aunque muchos días fueron difíciles siento que todas esas batallas no las peleé solo, siempre estaba ella como un pilar fundamental en mi vida.

A mi compañero de proyecto técnico Santiago que en toda la universidad hicimos proyectos juntos muchas veces se puso a liderar y en esta ocasión no fue la excepción tiene un gran potencial, sigue adelante y lucha por tus sueños

Por todo esto y mucho más les dedico este proyecto. Todos ustedes son el motor de mi vida y sin su apoyo nada de esto sería alcanzado.

Franklin Stalin Quiroga Lozano

## **DEDICATORIA**

Este proyecto se lo dedico a mi familia con todo mi amor y cariño ya que han sido las personas que siempre han buscado mi bienestar en todo ámbito.

A mis padres especialmente a mi madre que ha sido mi motor y apoyo en todos estos años de estudio, gracias a su amor, bendición y paciencia he logrado culminar esta etapa en mi vida profesional, a mis hermanos quienes con sus palabras de aliento me impulsaron a seguir adelante. A mis abuelitos que han sido un pilar fundamental durante toda mi carrera universitaria y con su experiencia me han dejado grandes enseñanzas como a ser perseverante y cumplir mis ideales. A mis tíos Iván y Gaby quienes me han brindado su apoyo incondicional, su motivación constante hizo que alcance esta meta tan importante para mí.

A mi compañero de tesis, Franklin que ha demostrado ser excelente ser humano y un gran amigo, ya que gracias a su desempeño, perseverancia y conocimiento hemos podido avanzar en cada desafío que se ha presentado en nuestra carrera universitaria.

Santiago Andrés Rosero Ibujés

## **AGRADECIMIENTO**

A nuestros padres por su constante comprensión, paciencia, trabajo y apoyo durante todo el tiempo de nuestra formación personal y profesional.

A la Universidad Politécnica Salesiana, por brindarnos los recursos necesarios para la elaboración del presente proyecto y por su excelente formación académica y humana.

Agradecemos a los docentes que a lo largo de estos años nos han ayudado en nuestra formación compartiendo sus conocimientos y enriqueciéndonos de habilidades que han sido fundamentales para nuestro desarrollo profesional.

Franklin Stalin Quiroga Lozano

Santiago Andrés Rosero Ibujés

## ÍNDICE

INTRODUCCIÓN .....	1
Antecedentes .....	1
Planteamiento del Problema.....	2
Justificación.....	2
Objetivos .....	3
Objetivo General .....	3
Objetivos Específicos.....	3
Marco metodológico .....	4
CAPÍTULO I.....	6
MARCO DE REFERENCIA .....	6
1.1 Marco Institucional .....	6
1.1.1 Organigrama de la carrera Ingeniería en Ciencias de la Computación .....	8
1.1.2 Flujo de Trabajo.....	8
1.2 Marco Teórico .....	9
1.2.1 Servicio de videoconferencia.....	9
1.2.2 Elementos básicos de videoconferencia .....	10
1.2.3 Arquitectura del sitio web (cliente-servidor).....	10
1.2.4 Códec de video VP9 .....	11
1.2.5 Herramientas de desarrollo tecnológico .....	11
1.2.6 Herramientas de desarrollo software .....	13
1.2.7 Diagramas UML.....	18
1.2.8 Metodología Xtreme Programming.....	18
1.2.9 Ciclo de vida de XP .....	18
CAPITULO II .....	20
ANÁLISIS Y DISEÑO .....	20
2.1 Especificaciones de requerimientos .....	20
2.1.1 Requerimientos funcionales.....	26
2.1.2 Requerimientos no funcionales.....	26
2.2 Historias de usuarios .....	27
2.3 Diagramas UML.....	34
2.3.1 Registro del usuario en el sistema .....	34
2.3.2 Login de usuario .....	36
2.3.3 Creación de Sala .....	37

2.3.4	Ingreso a la sala .....	39
2.1.1	Funcionalidades de la sala de videoconferencia.....	41
2.4	Análisis de Factibilidad.....	44
2.4.1	Factibilidad Económica .....	44
2.4.2	Factibilidad Técnica.....	47
2.4.3	Factibilidad operativa .....	48
2.4.4	Análisis costo beneficio.....	50
CAPITULO III .....		52
CONSTRUCCIÓN PLATAFORMA WEB DE VIDEOCONFERENCIA .....		52
3.1	Base de datos.....	52
3.1.1	Esquemas para la base de datos .....	52
3.2	Desarrollo e Implementación .....	53
3.2.1	Diagrama de Despliegue.....	53
3.2.2	Diagrama de navegación.....	54
3.3	Componentes del sitio web .....	55
3.4	Código importante.....	57
3.4.1	Módulo de Inicio, Registro y Acceso a la plataforma .....	58
3.4.2	Módulo de Videoconferencia .....	62
3.5	Dimensionamiento .....	71
3.6	Pruebas .....	73
3.6.1	Pruebas de caja negra.....	73
3.6.2	Pruebas de aceptación (UAT).....	81
3.6.3	Pruebas de carga .....	83
3.6.4	Pruebas de rendimiento .....	85
GLOSARIO.....		88
CONCLUSIONES .....		90
RECOMENDACIONES .....		92
LISTA DE REFERENCIAS .....		93



## ÍNDICE DE TABLAS

Tabla 1. Comparativa de metodologías ágiles .....	4
Tabla 2 Flujo de trabajo para una plataforma de videoconferencia .....	9
Tabla 3. Lista de etiquetas HTML.....	15
Tabla 4. Sentencias SQL .....	17
Tabla 5. Requerimientos funcionales .....	26
Tabla 6. Requerimientos no funcionales .....	26
Tabla 7. Historia de usuario Registro.....	27
Tabla 8. Historia de usuario Ingreso .....	28
Tabla 9. Historia de usuario Sistema de videoconferencia .....	29
Tabla 10. Historia de usuario Creación sala.....	30
Tabla 11. Historia de usuario Compartir archivos .....	31
Tabla 12. Historia de usuario Opciones .....	31
Tabla 13. Historia de usuario Mensajes .....	32
Tabla 14. Historia de usuario Grabar .....	32
Tabla 15. Historia de usuario Botón de regreso .....	33
Tabla 16. Especificaciones para el registro de usuario .....	34
Tabla 17. Especificaciones para el logeo .....	36
Tabla 18. Especificaciones para salas .....	37
Tabla 19. Especificaciones para ingreso a la sala .....	39
Tabla 20. Especificaciones de las funcionalidades de videoconferencia .....	41
Tabla 21. Costo en base a funcionalidades.....	45
Tabla 22. Costos de Hardware .....	46
Tabla 23. Hardware máquina virtual.....	47
Tabla 24. Software máquina virtual .....	47
Tabla 25. Comparativa cloud .....	48
Tabla 26. Comparativa entre soluciones de videoconferencia .....	51
Tabla 27. Campos de la base de datos.....	53
Tabla 28. Estructura de la plataforma de videoconferencia .....	55
Tabla 29. Prueba de caja negra para Registro .....	73
Tabla 30. Prueba de caja negra para acceder a la plataforma.....	74
Tabla 31. Prueba de caja negra para acceder a una sala.....	74
Tabla 32. Prueba de caja negra para crear una sala.....	76
Tabla 33. Prueba de caja negra para compartir archivos.....	77
Tabla 34. Prueba de caja negra para la interacción .....	78
Tabla 35. Prueba de caja negra para el chat .....	78
Tabla 36. Prueba de caja negra para grabar.....	79
Tabla 37. Prueba de caja negra de Logout .....	80
Tabla 38. Pruebas de aceptación .....	81
Tabla 39. Pruebas de carga de Register.....	83
Tabla 40. Pruebas de carga de Login .....	84
Tabla 41. Consumo de CPU por operación.....	87

## ÍNDICE DE FIGURAS

Figura 1. Organigrama de carrera de Ingeniería en Ciencias de la Computación.....	8
Figura 2. Requerimiento de herramienta de videoconferencia propia .....	20
Figura 3. Requerimiento para enviar mensajes .....	21
Figura 4. Requerimiento de usuarios activos .....	21
Figura 5. Requerimiento para silenciar micrófono.....	22
Figura 6. Requerimiento de tiempo en la plataforma.....	22
Figura 7. Requerimiento de un botón de participación .....	23
Figura 8. Requerimiento de modificar y eliminar usuarios.....	23
Figura 9. Requerimiento para intercambio de archivos .....	24
Figura 10. Requerimiento para grabar pantalla.....	24
Figura 11. Información navegadores.....	25
Figura 12. Información velocidad de internet .....	25
Figura 13. Diagrama Caso de uso registro de usuario.....	34
Figura 14. Diagrama de flujo para Registro de usuario .....	35
Figura 15. Diagrama de secuencia para registro de usuario.....	35
Figura 16. Diagrama de Caso de uso para acceso a la plataforma .....	36
Figura 17. Diagrama de flujo para acceder al sistema .....	36
Figura 18. Diagrama de secuencia de Login en el sistema .....	37
Figura 19. Diagrama Caso de uso para crear una sala de videoconferencia .....	38
Figura 20. Diagrama de flujo para crear una sala.....	38
Figura 21. Diagrama de secuencia para crear una sala.....	39
Figura 22. Diagrama Caso de uso para acceder a la sala de videoconferencia .....	40
Figura 23. Diagrama de flujo para acceder a una sala .....	40
Figura 24. Diagrama de secuencia para acceder a una sala de videoconferencia .....	41
Figura 25. Diagrama Caso de uso para participar en la sala .....	42
Figura 26. Diagrama de flujo para participar en la sala .....	43
Figura 27. Diagrama de secuencia para interactuar en la videoconferencia .....	44
Figura 28. Diagrama de despliegue de la plataforma de videoconferencia.....	53
Figura 29. Diagrama de navegación de la plataforma de videoconferencia .....	55
Figura 30. Estructura de componentes .....	57
Figura 31 Servidor web .....	58
Figura 32. Conexión a base de datos.....	59
Figura 33. Ruta archivos handlebars .....	59
Figura 34. Validación Login .....	61
Figura 35. Validación Register.....	62
Figura 36. Redireccionamiento .....	62
Figura 37. Creación de sala .....	63
Figura 38. Generar URL de la sala.....	63
Figura 39. Ingresar a la videoconferencia .....	64
Figura 40. Módulo.....	64
Figura 41. Estructura getUserMedia .....	65
Figura 42. Estructura shareScreen.....	65
Figura 43. Estructura saveRecordedStream .....	66
Figura 44. Mostrar archivos subidos al servidor .....	66

Figura 45. Subir archivos .....	67
Figura 46. Credenciales servidor turn .....	67
Figura 47. Diagrama de secuencia de websockets .....	68
Figura 48. Dependencia socket.io .....	69
Figura 49. Conexión socket.....	69
Figura 50. Estructura chat .....	70
Figura 51. Primeras peticiones para prueba en Register .....	83
Figura 52. Parámetros para la ejecución de la prueba en Register.....	84
Figura 53. Primeras peticiones para prueba en Login.....	85
Figura 54. Parámetros para la ejecución de la prueba en Login.....	85
Figura 55. Consumo de CPU.....	86
Figura 56. Consumo de disco.....	86
Figura 57. Consumo de red .....	87

## **RESUMEN**

El presente documento tiene como objetivo detallar la construcción de un sistema de videoconferencia que permita a los usuarios de la Universidad Politécnica Salesiana poder recibir clases de manera virtual. Esto con el propósito de brindar una alternativa cuando se presente algún problema que pueda interferir con el cronograma de actividades que ya se encuentra establecido por la universidad.

El desarrollo del programa fue realizado en base a los requerimientos de docentes y estudiantes. El sistema se encuentra alojado en una cloud, se utilizó un servidor Linux el cual se configuró para poder desarrollar el software, las configuraciones consistieron en la instalación de paquetes, dependencias, puertos de comunicación y protocolos.

La estructura del documento se la detalla de la siguiente manera: en el capítulo 1 se encuentra toda la información referente a la universidad y los conceptos teóricos que se han utilizado para el desarrollo del software, en el capítulo 2 se puede encontrar el análisis realizado para definir los requerimientos del proyecto, el diseño en base a los diagramas que permitirán definir el comportamiento del sistema y los análisis de factibilidad para conocer si el proyecto es viable, y finalmente el capítulo 3 tiene toda la información relevante del código junto con las pruebas ejecutados en donde se puede verificar que el proyecto puede ser una alternativa ante otras soluciones.

## **ABSTRACT**

The objective of this document is to detail the construction of a videoconferencing system that allows users of the Salesian Polytechnic University to receive classes virtually. This in order to provide an alternative when there is a problem that may interfere with the schedule of activities that is already established by the university.

The development of the program was carried out based on the requirements of teachers and students. The system is hosted in a cloud, a Linux server was used which was configured to be able to develop the software, the configurations consisted of the installation of packages, dependencies, communication ports and protocols.

The structure of the document is detailed as follows: in chapter 1 you will find all the information regarding the university and the theoretical concepts that have been used for the development of the software, in chapter 2 you can find the analysis carried out for define the project requirements, the design based on the diagrams that will allow to define the behavior of the system and the feasibility analyzes to know if the project is viable, and finally chapter 3 has all the relevant information of the code together with the tests executed where it can be verified that the project can be an alternative to other solutions.

## INTRODUCCIÓN

### Antecedentes

Actualmente la comunicación es una necesidad en la vida cotidiana, y esto ha incrementado el uso de herramientas tecnológicas permitiendo optimizar, agilizar y mejorar las actividades que se realizan día a día. Sin embargo, el campo de la educación carece de plataformas que permitan una comunicación bidireccional entre docente y alumno.

Las principales herramientas de videoconferencia no se enfocan en el aprendizaje, por tal motivo la creación de un software dedicado es indispensable en el ámbito educativo. La mayoría de software de videoconferencia cuenta con varias restricciones, por ejemplo, “Skype es un software que permite que todo el mundo se comuniquen. Millones de personas y empresas ya usan Skype para hacer llamadas y videoconferencias gratis” (Microsoft, 2019), un inconveniente en el uso de esta herramienta es que solo se puede acceder si se tiene una cuenta en Outlook. En el caso de Zoom, la suscripción es de 49 dólares por mes, mientras que el plan gratuito tiene varias limitantes, permite realizar videollamadas con un límite de 100 participantes, con un mínimo de 40 minutos. La instalación y funcionamiento de estos programas dependen mucho de los recursos del ordenador

Una alternativa que surgió fue la telefonía IP o software de videoconferencia, pero esta presenta varias deficiencias como: uso excesivo de ancho de banda, problemas con la conectividad, dificultad en su configuración. Actualmente existen nuevas herramientas que podrían ayudar en la comunicación, como la tecnología Web Real Time Communication (Constantin, Calinciuc, Octavian, & Filote, 2016).

## **Planteamiento del Problema**

Los programas para videoconferencia presentan un gran problema de seguridad para los usuarios ya que las videoconferencias pueden ser espiadas por terceros, las credenciales de los usuarios son filtrados en sitios web no indexados. Como solución a estos problemas WebRTC ha llegado para mejorar la comunicación en la transmisión de audio y video, por otro lado, es de código abierto, permite conexiones peer-to-peer y soporta varios navegadores que al contar con certificaciones SSL garantiza la seguridad en la capa de transporte (Moaid, Al-Sherbaz, & Turner, 2017).

La Universidad Politécnica Salesiana sede Quito campus Sur cuenta con la plataforma virtual AVAC (Ambiente Virtual de Aprendizaje Cooperativo) la cual permite subir el material de estudio por parte de los docentes y material de retroalimentación de los alumnos, un inconveniente identificado es que carece de una herramienta de videoconferencia propia, la misma que permita continuar con el cronograma de actividades y el plan analítico establecido.

Implementar un sitio web de videoconferencia propio mejorará la comunicación entre el docente y alumno, mediante las comunicaciones bidireccionales se podrán obtener interacciones más complejas, como preguntas en tiempo real e intercambio de contenido (Oh, Ahn, Choi, & Yang, 2015).

## **Justificación**

Implementar una tecnología de comunicación que permita la conexión simultánea entre dos o más personas que geográficamente se encuentre alejados para transmitir información. Estas herramientas logran incorporar múltiples recursos externos, teniendo como objetivo compartir documentos y posibilitando reuniones evitando costos de desplazamiento. “Los rápidos progresos de la tecnología de la información y la comunicación modifican la forma de elaboración, adquisición y transmisión de conocimientos” (UNESCO, 1996).

La educación actual debe mejorar las herramientas de formación académica, garantizando los procesos de enseñanza. E-learning puede ofrecer las herramientas tecnológicas necesarias para la gestión de procesos E/A, el cual puede ser implementado en un sistema presencial mediante un cambio de estrategias tradicionales, de manera que se puedan complementar las ventajas de interacción cara a cara con las potencialidades de la red, como también puede emplearse para potenciar los aprendizajes en un entorno virtual o una combinación de ambas, por esta razón, e-learning no se concibe como educación a distancia, sino como presencial diferida en tiempo y el espacio (Morales, 2010).

Incorporar la aplicación web de videoconferencia tiene como finalidad facilitar el uso de salas múltiples para que los docentes puedan impartir sus clases, brindando a la universidad una solución ante posibles irregularidades. Usar un servidor permitirá sincronizar las conexiones entre distintos nodos.

## **Objetivos**

### **Objetivo General**

Analizar, diseñar, y construir una Aplicación de videoconferencia utilizando el protocolo Web Real-Time Communication, para impartir clases virtuales a los estudiantes de la carrera de Ingeniería en Sistemas y Ciencias de la Computación de la Universidad Politécnica Salesiana sede Quito campus Sur.

### **Objetivos Específicos**

Analizar las características mínimas de videoconferencia mediante la recolección de datos que permitan identificar las necesidades del estudiante, docentes y Administrador

Diseñar diagramas UML para el desarrollo de la aplicación web mediante el uso de metodología XP para solventar los requerimientos del usuario.



Construir una aplicación web que permita la interacción entre el docente y los estudiantes de la carrera de Ingeniería de Sistemas y Computación

Dimensionar la infraestructura que permitirá dar soporte al funcionamiento adecuado de la aplicación cumpliendo los requerimientos de los usuarios.

Preparar las pruebas respectivas que permitan evaluar la satisfacción del usuario con respecto al sistema por medio de la recopilación de datos

### Marco metodológico

Tabla comparativa de las metodologías que se podrían adaptar al desarrollo de la aplicación web.

Tabla 1. Comparativa de metodologías ágiles

<b>Características</b>	<b>XP</b>	<b>Scrum</b>	<b>Kanban</b>	<b>Rup</b>
Trabajo en grupo de dos	X			
Se enfoca en la creación del producto	X			
Modifica el orden de las tareas de desarrollo		X		
Metodología basada en prueba y error	X			
Requiere por lo menos una reunión semanal	X	X		
Requiere de resultados de manera inmediata		X		
Rendimiento limitado			X	
Enfocada en la gestión de proyectos		X	X	
Administra el ambiente de desarrollo	X			X
Guarda todas las versiones del proyecto				X

Nota: Son las características de cada metodología ágil

La metodología que se ajusta de mejor manera según sus características es XP, es ideal para el desarrollo de una aplicación que se encuentra en constante retroalimentación, administrar los ambientes de desarrollo es fundamental para este tipo de proyectos donde se puede experimentar diferentes tecnologías(servidores) y lenguajes de programación, para verificar el correcto funcionamiento en las diferentes etapas de programación se realizan pruebas para encontrar y corregir errores que puedan afectar la ejecución del software. Por ende, esta metodología se enfoca en la creación del producto independientemente de las tecnologías de desarrollo que se utilicen. El trabajo en grupo es indispensable, los dos participantes tienen sus tareas asignadas y en las reuniones semanales se verificará los avances respectivos para tener una visión general del proyecto y poder identificar que etapas deben tener mayor prioridad.

# CAPÍTULO I

## MARCO DE REFERENCIA

### 1.1 Marco Institucional

El 5 de agosto de 1994, el presidente de la República, Arquitecto Sixto Durán Ballén, firma el decreto presidencial de creación de la Universidad Politécnica Salesiana del Ecuador. La UPS nace en una época muy crítica desde el punto de vista social y económico.

Una vez aprobado el proyecto de creación de nuestra Universidad, la Sociedad Salesiana del Ecuador inicia las actividades del nuevo Centro de Educación Superior en octubre de 1994. Previamente, el 6 de septiembre de 1994 se instala el primer Consejo Universitario y se realiza la posesión del Rector y Vicerrector. La Universidad Politécnica Salesiana, como centro de educación superior, es consciente de los grandes problemas educativos que afronta el país tales como:

- La necesidad de formar un profesional integral, científico, práctico, humano, moral y ético.
- La necesidad de vinculación de la universidad con la sociedad.
- La necesidad de que la ciencia y la tecnología sean parte de un mundo integrador de la formación.
- La necesidad de que la investigación esté vinculada a la solución de los grandes problemas sociales.

Para la carrera de Ingeniería en Ciencias de la Computación es importante formar Ingenieros e Ingenieras en Computación con una sólida base de conocimiento técnico y científico sustentada en un enfoque humanista y de responsabilidad ética y social, capaces de liderar innovar y desarrollar respuestas tecnológicas que permitan contribuir a la búsqueda de soluciones para

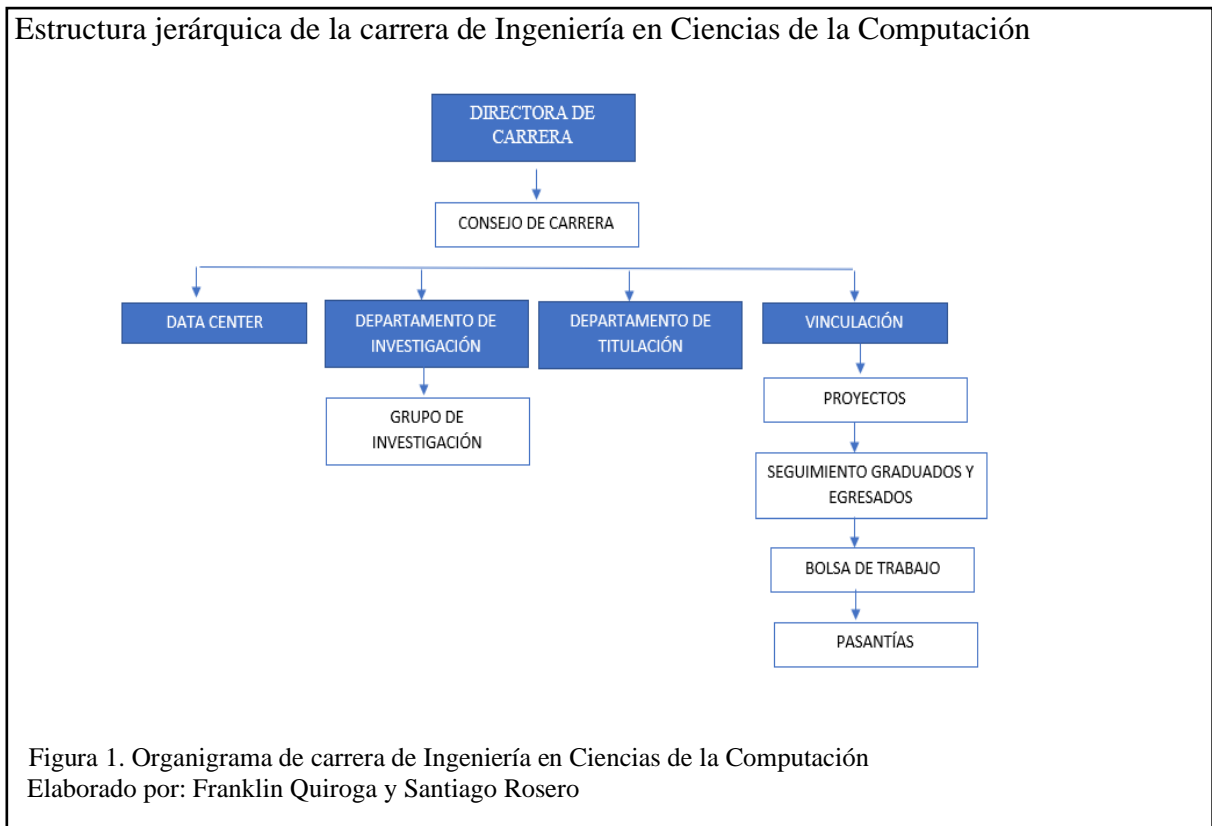
problemas de la sociedad y el desarrollo del país, alineado a estándares de calidad nacionales e internacionales.

Uno de los principales retos que afrontó la Universidad Politécnica Salesiana en Octubre 2019 fue mantener la continuidad del plan analítico de los estudiantes de las diferentes carreras, las manifestaciones que duraron alrededor de dos semanas tuvieron un alto impacto que produjo que muchas de las carreras suspendieran sus actividades, esto trajo como consecuencia que el cronograma de estudios no sea cumplido de la manera establecida dando como resultado que muchos de los docentes opten por dar clases a través de software de videoconferencia. Si bien la Universidad no tiene un plan de contingencia ante este tipo de eventualidades es indispensable que se considere tener una herramienta de videoconferencia propia que pueda solventar estos acontecimientos.

Con lo sucedido en todo el 2020 por la emergencia sanitaria a nivel mundial, la Universidad Politécnica Salesiana se ha visto en la necesidad de adquirir una solución de videoconferencia para impartir clases al alumnado. Estas soluciones representan un costo elevado que tal vez no atiende a todas las necesidades que se presentan sin mencionar los severos problemas de seguridad informática que estos tienen como son la filtración de correos electrónicos y fotos, cifrado en la comunicación y falsos instaladores de software que por lo general tiene incorporado malware.

En vista de toda la problemática que sufre el país y por lo tanto el sistema educativo, es necesario elaborar un sistema de videoconferencia propia que sirva como solución y se adapte a las necesidades de los usuarios, ya que esta propuesta de videoconferencia permitirá incorporar funcionalidades que el resto de herramientas no cuenta. Este desafío se alinea a los principios de la UPS en formar estudiantes que desarrollen tecnología para solucionar las problemáticas de la sociedad.

### 1.1.1 Organigrama de la carrera Ingeniería en Ciencias de la Computación

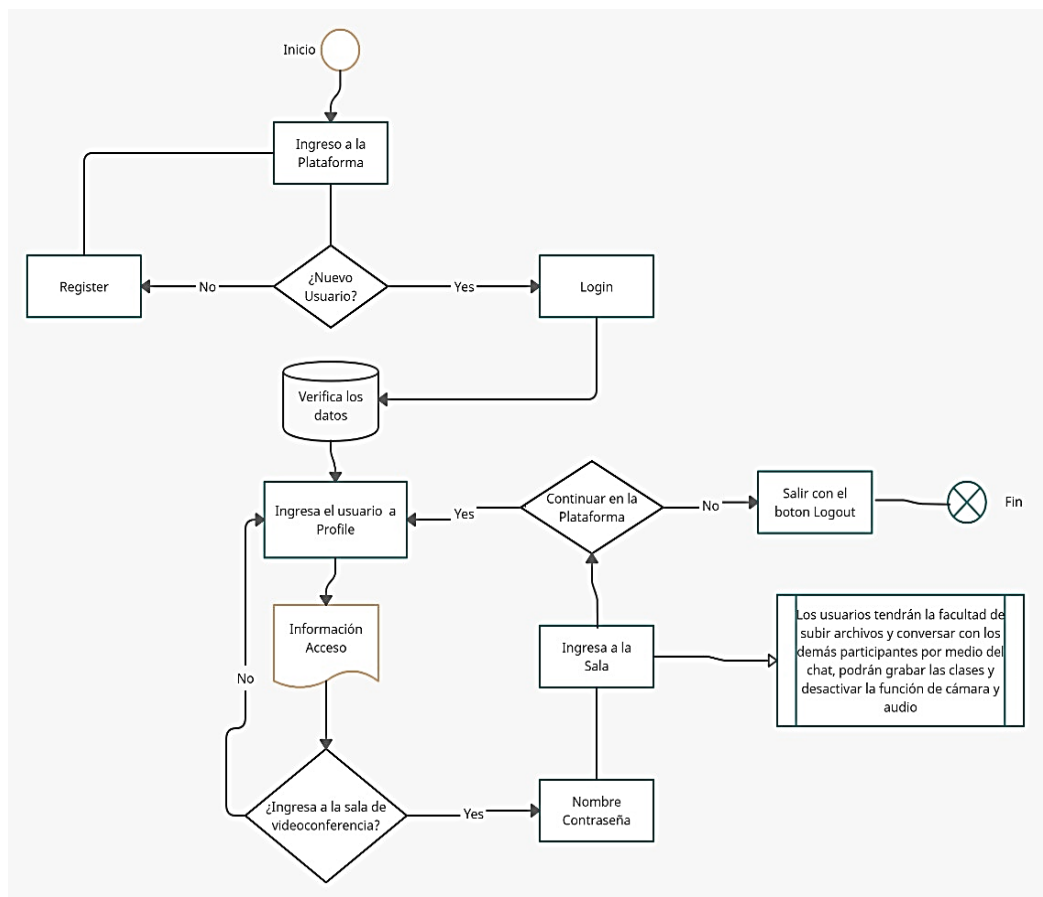


### 1.1.2 Flujo de Trabajo

En la Tabla 2 se mostrará el flujo que se realiza para que un docente o estudiante pueda ingresar al sistema de videoconferencia, en donde primero se valida que cada usuario ingrese su correo, contraseña y rol. El siguiente proceso lo realiza al verificar que se ingrese el nombre de usuario y contraseña de la sala, si los datos son correctos el sistema proporcionara el acceso a una sala de videoconferencia en la cual se tiene las diferentes funcionalidades que ofrece un software de videoconferencia. En caso de ser un usuario que no se encuentra registrado, el podrá registrarse en la plataforma para tener acceso al sistema. Por último, se tiene la opción de salir de la sala de a través del botón Logout.

Tabla 2 Flujo de trabajo para una plataforma de videoconferencia

<p><b>Descripción:</b> Proceso por el que los usuarios (docentes y alumnos) ingresa a la plataforma de videoconferencia para realizar diferentes actividades</p>	<p><b>Actividad:</b> Ingresar a la plataforma de</p>
	<p><b>Subactividades:</b> Compartir archivos, interactuar con los demás participantes por medio del chat</p>
	<p><b>Involucrados:</b> Administrador, Docentes, Alumnos</p>



Nota: Proceso para el ingreso a la videoconferencia

## 1.2 Marco Teórico

### 1.2.1 Servicio de videoconferencia

Un servicio que ha crecido en estos últimos años se trata de videoconferencias, este utiliza medios que permiten compartir información a través de programas o dispositivos, las

instituciones educativas que más lo utilizan son universidades debido a la factibilidad que se tiene al transmitir audio y video por un mismo canal. Dentro de la educación la videoconferencia puede solventar funciones específicas en los procesos de aprendizaje y fomentar la investigación (Sánchez Arroyo, 2001).

Una de las principales características de videoconferencia es poder realizar una conexión bidireccional, tener un control en los equipos de audio y video que permitan entablar una comunicación en tiempo real (Medina, 2003).

### **1.2.2 Elementos básicos de videoconferencia**

Existen 3 elementos importantes de la videoconferencia que son: Sala de conferencia es la plataforma en donde un usuario accede mediante su dispositivo para interactuar con los demás participantes que se encuentren en la misma sala, a su vez, podrá acceder a los controles del equipo de audio y video que esté utilizando, el nivel de confort que permita al usuario sentirse a gusto en el software y por último el codec el cual permitirá codificar-decodificar los datos obtenidos entre los extremos (Morales Salcedo & others, 1999).

### **1.2.3 Arquitectura del sitio web (cliente-servidor)**

Un sitio web de videoconferencia está basado en una arquitectura cliente servidor, este permite repartir los recursos o servicios que pueden estar en una misma máquina o en diferentes. Los clientes son aquellos que realizan la petición mientras que el servidor es el encargado de responder dichas demandas. En esta arquitectura se tiene una distribución de la información centralizada. En este modelo se puede destacar lo siguiente: no es necesario contar con servidor de aplicación ya que utiliza en la infraestructura los servicios de virtualización (Marini, 2012).

El comportamiento habitual de un cliente es realizar varias peticiones al servidor, las más comunes que se suelen registrar son: identidad de usuario, contraseña de acceso, fecha y hora

de petición, rol de usuario y el path. Esta información ayuda a la toma de decisiones en las mejoras continuas del sitio web (Hassan et al., 2004).

#### **1.2.4 Códec de video VP9**

El estándar de compresión VP9 es el sucesor de VP8 desarrollado por Google con el objetivo de satisfacer la demanda de consumo de video en internet, incluido el contenido de video a pedido y producido tanto por profesionales y aficionados. Este formato de compresión de video open source hace los pixeles más grandes y a su vez reduce la tasa de bits (por lo tanto, ocupará menos espacio) y como resultado se obtiene un video con el doble de resolución (Shahriar, 2014).

#### **1.2.5 Herramientas de desarrollo tecnológico**

##### ***1.2.5.1 Cloud***

La computación en la nube se enfoca en concentrar las funcionalidades de una computadora y su almacenamiento en un núcleo, el cual cuenta con características capaces de elevar el rendimiento y mejoran la accesibilidad hacia los servicios. Los sistemas operativos en la nube tienen como función ejecutar aplicaciones que continúen funcionando incluso si la cantidad de usuarios aumenta (Hayes, 2008).

##### ***1.2.5.2 Máquina virtual***

Una máquina virtual es un simulador de hardware en el cual se instala un sistema operativo para su funcionamiento. Este software se encuentra en servidores con diferentes características (Shieh, 2013).

El termino instancia se refiere a una máquina virtual que se aloja en una infraestructura computacional, para realizar dicha instancia en una de las diferentes empresas que ofrecen este servicio es hacerlo mediante línea de comandos o en ciertos casos con el uso de una API. Cada



proyecto que se ejecute sobre una instancia debe contar con un sistema operativo, una opción de almacenamiento y una instancia de red.

Lo más relevante que se puede relacionar: se puede migrar de manera activa sin necesidad de hacer un reinicio al sistema, la gestión de parches ayuda a mejorar la seguridad del sistema operativo (Google, 2020).

### ***1.2.5.3 Ubuntu***

Es un sistema operativo basado en Linux que fue lanzado en el 2004 que tiene la misión de reducir los costos de los servicios, a su vez entregar un software libre de calidad a todas partes del mundo. Ubuntu se encuentra tanto en ordenadores de escritorio hasta nubes informáticas.

Para la versión 18.04 se han mejorado características como: la encriptación de memoria AMD, mejoró el soporte para el hardware, controlar el uso de la energía y la interfaz de CPU (Ubuntu, 2020).

### ***1.2.5.4 Filezilla***

Es un software de código abierto que utiliza los protocolos de transferencia de archivo FTP, FTPS y SFTP para gestionar, transferir y acceder a archivos de manera estable y segura mediante los protocolos de seguridad SSL y SSH. También trabaja con una arquitectura cliente servidor el cual permitirá administrar archivos de sitio web desde cualquier ordenador independientemente del sistema operativo que utilice ya sea Windows, Mac o Linux (Woodraska et al., 2011).

## **Características**

- Soporta IPv6
- Soporta varios idiomas
- Admite reanudar transferencias y archivos inconclusos
- El límite de velocidad de transferencia puede ser configurable

- Permite la edición y búsqueda remota de archivos (FileZilla, 2020).

#### **1.2.5.5 Atom**

Es un editor para el desarrollo web que soporta varios lenguajes de programación, entre lo más destacable se puede mencionar que dispone de todas las APIs típicas de una aplicación de Nodejs. Proporciona una experiencia de desarrollo híbrida por parte del cliente. Entre sus actualizaciones más significativas está en ignorar problemas de compatibilidad entre los navegadores.

#### **Características**

- Compatibilidad en múltiples plataformas
- Integración con Github
- Personalización del entorno
- Funcionalidades extra con los “package manager” (ATOM, 2020).

#### **1.2.6 Herramientas de desarrollo software**

##### **1.2.6.1 Javascript**

Es un lenguaje de programación interpretado que permite crear página web dinámicas, los programas escritos por javascript puede ser ejecutados en cualquier navegador que incluya soporte javascript los cuales con (Firefox, Opera, Safari, Chrome) como herramienta este se puede adaptar a varios lenguajes lo que mejora el flujo de la información (Eguéiluz Pérez, 2012).

#### **Características**

- **Es un lenguaje de guiones:** Javascript al no cumplir con todas las necesidades del cliente, permite que estos realicen modificaciones, para ello es importante que dichas instrucciones sean incluidas en el documento HTML para lo cual se utilizan scripts (también llamados guiones).

- **Basado en objetos:** Esta característica permite reaccionar a diferentes eventos (carga y descarga de páginas y envío de formularios) que tiene una página web, por lo cual es necesario configurar dichas acciones en los scripts.
- **Independiente de la plataforma:** Cualquier script puede ser ejecutado en cualquier navegador, su limitación se ve reflejada en la versión que este maneje (Maza, 2012).

#### *1.2.6.2 Nodejs*

Es un software que permite construir aplicaciones en JavaScript, se basa en un modelo de entrada y salida que resulta ser eficiente para el control de eventos. Su plataforma consta de 3 capas: la inferior contiene los componentes centrales, la capa media actúa como middleware y la capa superior contiene los APIs de JavaScript. Ofrece un gran conjunto de paquetes que son administrados por “npm”.

#### **Características**

- **Es asíncrono:** Permite la construcción de páginas web altamente escalables.
- **Motor Fast Script:** Cuenta con una gran velocidad de ejecución del lado del servidor.
- **Soporta MySQL:** Cuenta con interacciones y ajustes nativos con varios gestores de bases de datos (Js & JS, 2016).

#### *1.2.6.3 Librería socket.io*

Es una librería open source que facilita el desarrollo de aplicaciones mediante el uso de websockets tanto en el cliente como en el servidor. Una de las ventajas de esta es que se lo puede usar en diferentes lenguajes de programación. De la misma manera permite controlar varios eventos en tiempo real para que las aplicaciones puedan ser ejecutados en navegadores (Azaustre, 2016).

#### 1.2.6.4 *Html*

Es un lenguaje de etiquetas el cual permite crear páginas web extremadamente flexibles, en cuanto a diversidad de elementos disponibles se tiene HTML, CSS y JavaScript que trabajan conjuntamente como una sola aplicación, para su ejecución es necesaria que las instrucciones sean especificadas en HTML. Su estructura es organizada y es diferenciada por el uso de etiquetas específicas <html>: permite declarar el tipo de documento, <head>: muestra la información del archivo, <title>: Especifica el título del documento, <body>: Genera la parte visible de la página web (Gauchat, 2012).

#### **Características**

- Se puede integrar estilos que resulten ser más amigables con el entorno del usuario final.
- Cuenta con gran cantidad de funcionalidades que antes solo se lograban con el uso de plugins integrados en el código.
- Permite incorporar archivos de multimedia más complejas, entre las cuales se tiene imágenes, audio y video (Ríos, 2012).

Tabla 3. Lista de etiquetas HTML

<b>ETIQUETA</b>	<b>DESCRIPCIÓN</b>	<b>CATEGORIA</b>
<!-- ... -->	Comentario	Preámbulo
Link	Enlace a archivos	Metadatos
Style	Hoja de estilo del documento	Metadato
h1 a h6	Encabezado	Secciones
P	Párrafo	Bloques de contenidos
hr	Separador	Bloques de contenidos
div	División	Bloques de contenidos
br	Salto de línea	Texto
a	Hiperenlace	Texto
Span	Contenedor de texto genérico	Texto
Img	Imagen	Contenido incrustado

Source	Origen de imagen, audio y video	Contenido incrustado
Input	Control	Formularios
Button	Botón	Formularios
Option	Opciones en caja de lista	Formularios
Script	Elemento de JavaScript	Scripts

Nota: Son las etiquetas usadas para el desarrollo web

### **1.2.6.5 *Css***

Es un lenguaje de hoja de estilo que es utilizado para mejorar la presentación visual de la página web elaborada con HTML. Esta herramienta permite definir acciones complejas en el diseño.

Para vincular el Css al Html es necesario especificar en el encabezado la ruta del archivo Css en la etiqueta <style> (Stark et al., 2010).

### **1.2.6.6 *Base de datos***

Es una estructura que permite agrupar, almacenar y organizar los datos de una aplicación de manera que su búsqueda sea más rápida y eficaz. Las bases de datos resultan ser un eje importante en la administración de la información, al usar una base de datos dinámica se puede realizar operaciones como crear, actualizar, modificar y eliminar los datos (Díaz, 2015).

### **Características**

- Crea ambientes para compartir los datos de mejor manera.
- Mejora la seguridad de los datos restringiendo el acceso a usuarios no deseados.
- Permite la integración de los datos para mejorar la organización y visualización.
- Minimiza la inconsistencia de datos para controlar la data ante posibles errores (Coronel et al., 2011).

### **1.2.6.7 *Mysql***

Es un sistema gestor de base de datos (SGDB) utilizado ampliamente en distribuciones GNU Linux por su simplicidad y alto rendimiento. Este puede recibir peticiones del cliente a través

de TCP/IP, sockets. Para que el cliente pueda interactuar con la base de datos es necesario el uso de sentencias SQL las cuales permitirán hacer consultas, modificaciones e incluso eliminar registros que se encuentran en una tabla dentro de la base de datos.

Entre sus principales características se puede identificar las siguientes: su administración se basa en usuarios y privilegios, es altamente confiable en cuanto a estabilidad, utiliza arquitectura cliente servidor que es incrustado en aplicaciones por lo cual tiene rápida velocidad de respuesta (Santillán et al., 2014).

Las sentencias SQL que se utilizan para intervenir en una base de datos se los detalla en la Tabla 4.

Tabla 4. Sentencias SQL

<b>SENTENCIA</b>	<b>DESCRIPCIÓN</b>	<b>CATEGORIA</b>
Create	Crear una nueva base de datos o una nueva tabla	Lenguaje de definición de datos (DDL)
Insert	Insertar un registro a una tabla	Lenguaje de manipulación de datos (DML)
Update	Actualizar los registros de una tabla	Lenguaje de manipulación de datos (DML)
Alter	Actualización en las tablas de la base de datos	Lenguaje de definición de datos (DDL)
Drop	Eliminar toda la tabla de a base de datos	Lenguaje de definición de datos (DDL)
Delete	Eliminar un registro de una tabla	Lenguaje de manipulación de datos (DML)
Select	Consulta uno o varios registros de una tabla que satisfaga un criterio específico.	Lenguaje de manipulación de datos (DML)

Nota: Sentencias para la administración de base de datos

### **1.2.7 Diagramas UML**

Es un lenguaje universal de modelado el cual permite diseñar y evidenciar los elementos que conforman un software, consiste en documentar los detalles en la etapa de desarrollo en cada uno de los procesos. A su vez esta propone diferentes soluciones desde varios puntos de vista con el objetivo de implementar la mejor alternativa para la construcción de la plataforma (Grau & Segura, 2008).

### **1.2.8 Metodología Xtreme Programming**

Es una metodología ágil en el desarrollo de software, lo que convierte a los programas en algo mucho más fáciles de comprender ya que existe una mayor comunicación entre las personas. Se trabaja de manera continua con el cliente lo que facilita con la comunicación y la comprensión de los objetivos establecidos. XP permite realizar cambios en la toma de decisiones del desarrollo, esto puede implicar un alto costo en el proyecto, pero el tiempo de ejecución se mantiene de acuerdo a la fecha establecida, la calidad seguirá siendo la misma y el alcance del producto cumplirá con las funcionalidades asignadas por parte del usuario (Fernández, 2013).

### **1.2.9 Ciclo de vida de XP**

#### ***1.2.9.1 Planeación***

Es la etapa inicial en la cual el grupo de trabajo realiza su primera interacción con el cliente, aquí se establece el cronograma de actividades, levantamiento de requerimientos del sistema. Este punto es fundamental debido a que se pueden definir varios elementos importantes como son: velocidad del proyecto, número de iteraciones, avances del proyecto, actividades del personal y plan de entrega (Tobón & Carmona, 2007).

### ***1.2.9.2 Diseño***

Su objetivo de disminuir tiempo y esfuerzo al momento de implementar las diversas soluciones. Spike pretende abordar los problemas concretos y aislar otro tipo de preocupaciones, una cualidad a destacar es su simplicidad ya que intenta mejorar la comprensión del proyecto a todos los participantes sin necesidad de dar explicaciones complejas, su filosofía es mantener el código lo más simple posible y que su funcionalidad sea la esperada (Pérez Pérez & others, 2012).

### ***1.2.9.3 Desarrollo***

Es indispensable involucrar al cliente como parte del grupo de trabajo, el cliente proporcionará los detalles, durante la etapa del desarrollo los programadores discutirán las implementaciones y los estándares que el código debe cumplir. Estas actividades ayudaran a prevenir situaciones no deseadas (Joskowicz, 2008).

### ***1.2.9.4 Pruebas***

Todos los módulos de programación deben ser sometidos a una prueba antes a su entrega. Cada prueba debe ser respectivamente documentada junto al código, con el fin de corregir y cambiar partes del código sin alterar las distintas funcionalidades de la aplicación (Pérez Pérez & others, 2012).



## CAPITULO II

### ANÁLISIS Y DISEÑO

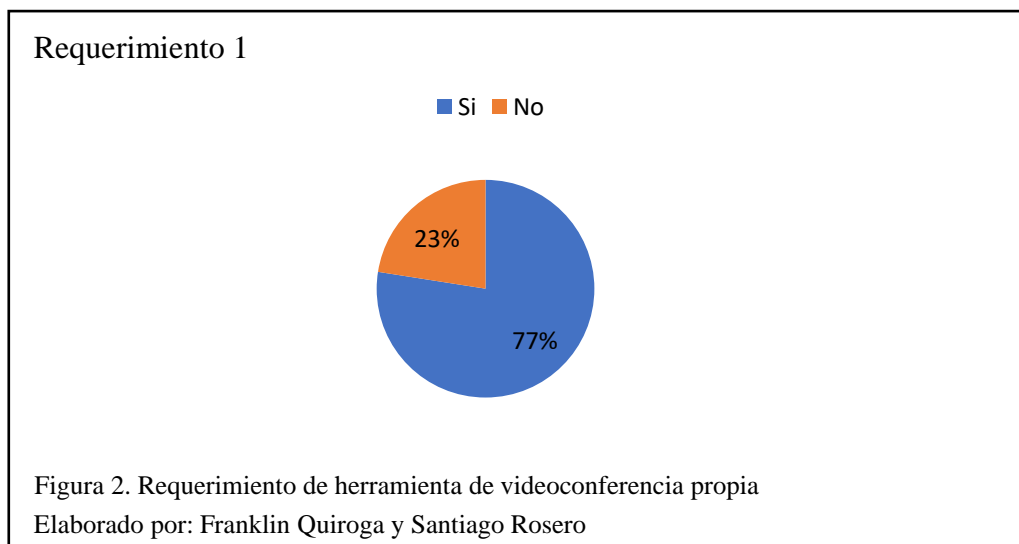
#### 2.1 Especificaciones de requerimientos

Para el levantamiento de requerimientos se estableció una encuesta orientada a obtener información relevante sobre el funcionamiento, las necesidades y que características debería tener el sistema de videoconferencia para la carrera de Ingeniería en Sistemas e Ingeniería en Ciencias de la Computación, obteniendo los siguientes resultados.

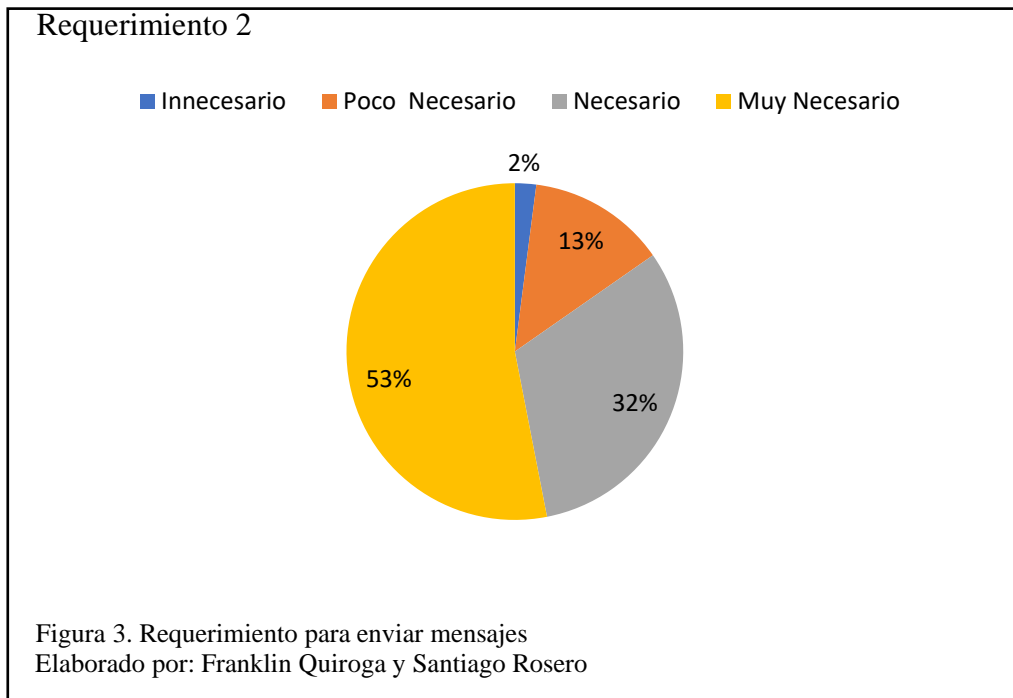
Para la tabulación de los datos se ha optado por unificar los resultados obtenidos en las encuestas de docentes y estudiantes.

De una total de 605 personas entre docentes y estudiantes de la carrera se contó con la participación de 111 que equivale al 18,34%. Para el análisis de requerimientos se optó por formular 12 preguntas, las mismas que tienen como objetivo determinar las funcionalidades básicas para el sistema de videoconferencia, sin embargo, en un enunciado se solicitó a las personas encuestadas escribir un requerimiento que se desee incorporar a la plataforma, obteniendo los siguientes resultados.

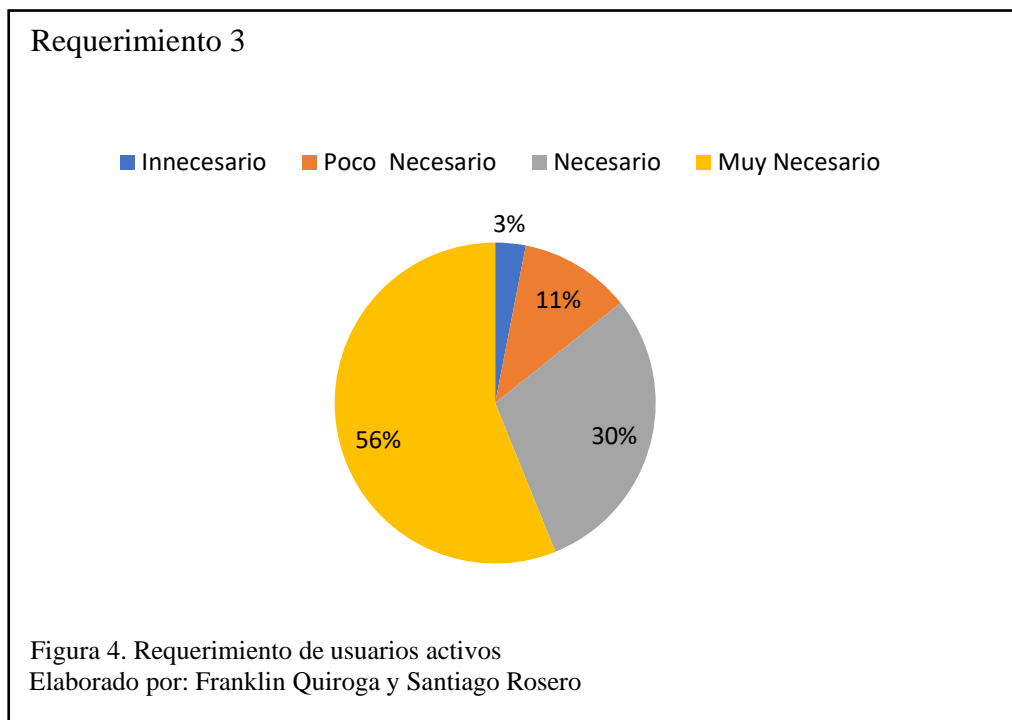
1. ¿Cree que es indispensable contar con una herramienta de videoconferencia propia?



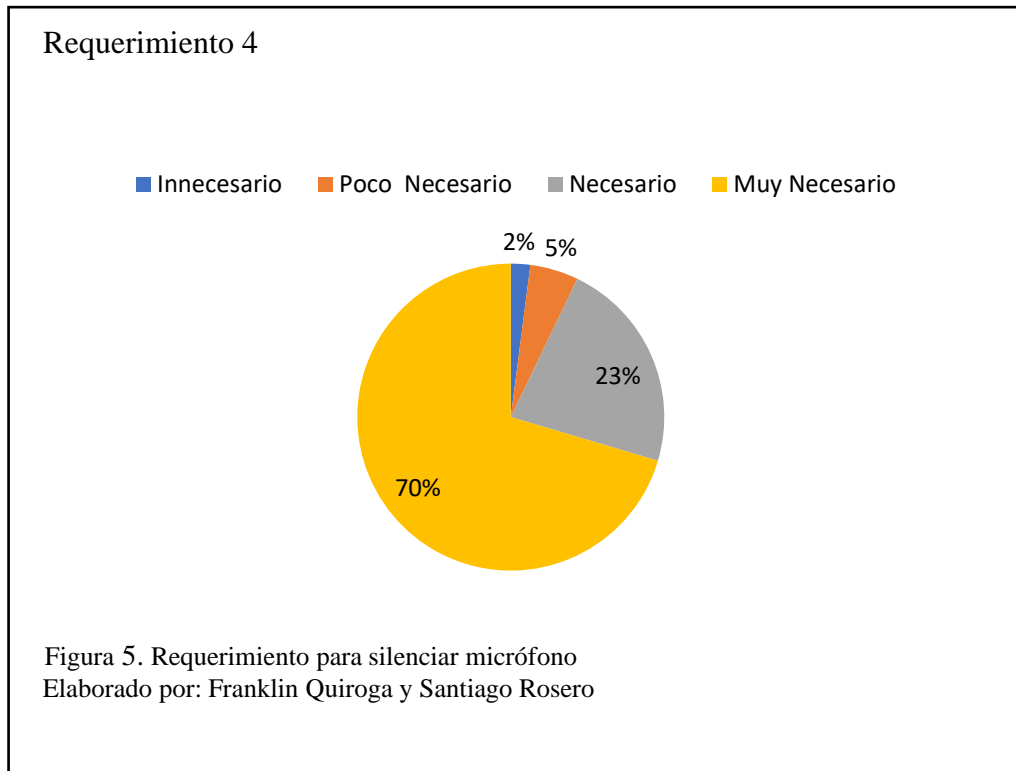
2. ¿Poder enviar mensajes privados?



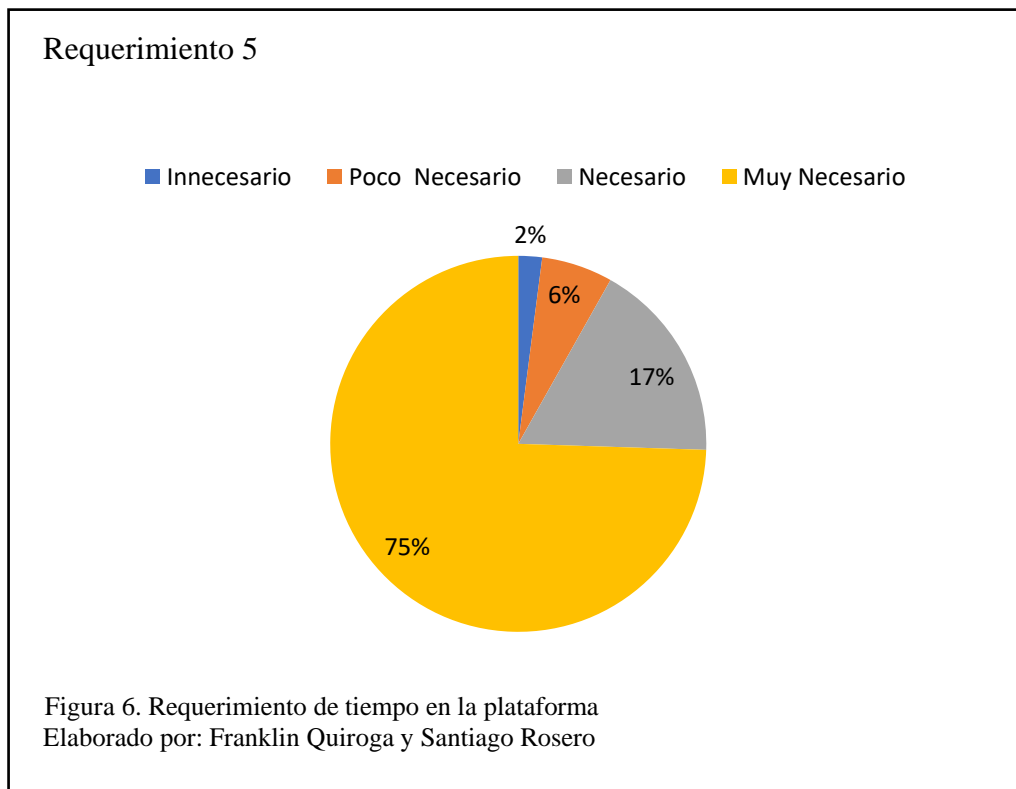
3. ¿Conocer qué usuarios se encuentran activos en la plataforma de videoconferencia?



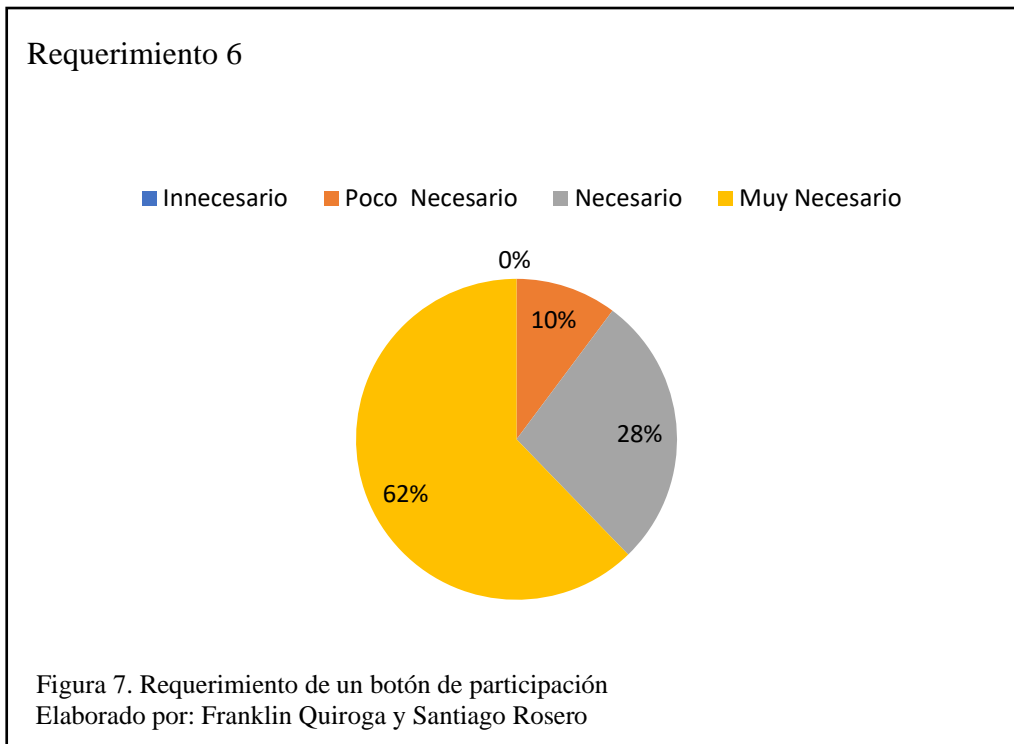
#### 4. ¿Silenciar el micrófono?



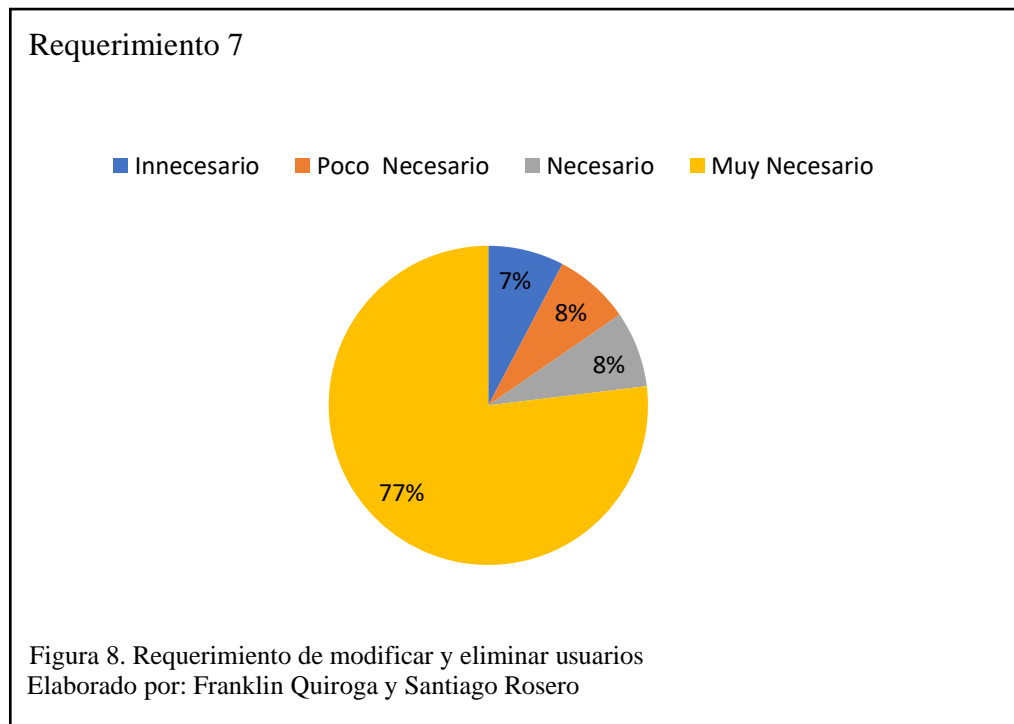
#### 5. ¿Tener tiempo indefinido de videoconferencia?



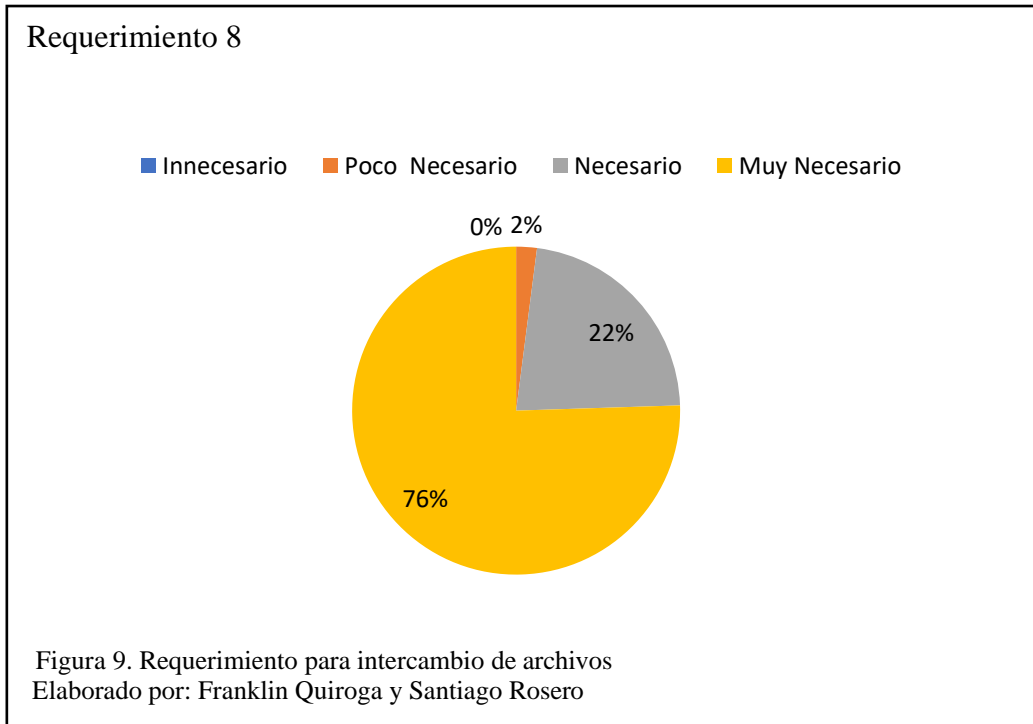
6. ¿Incorporar un botón, el mismo que indique que el estudiante desea participar?



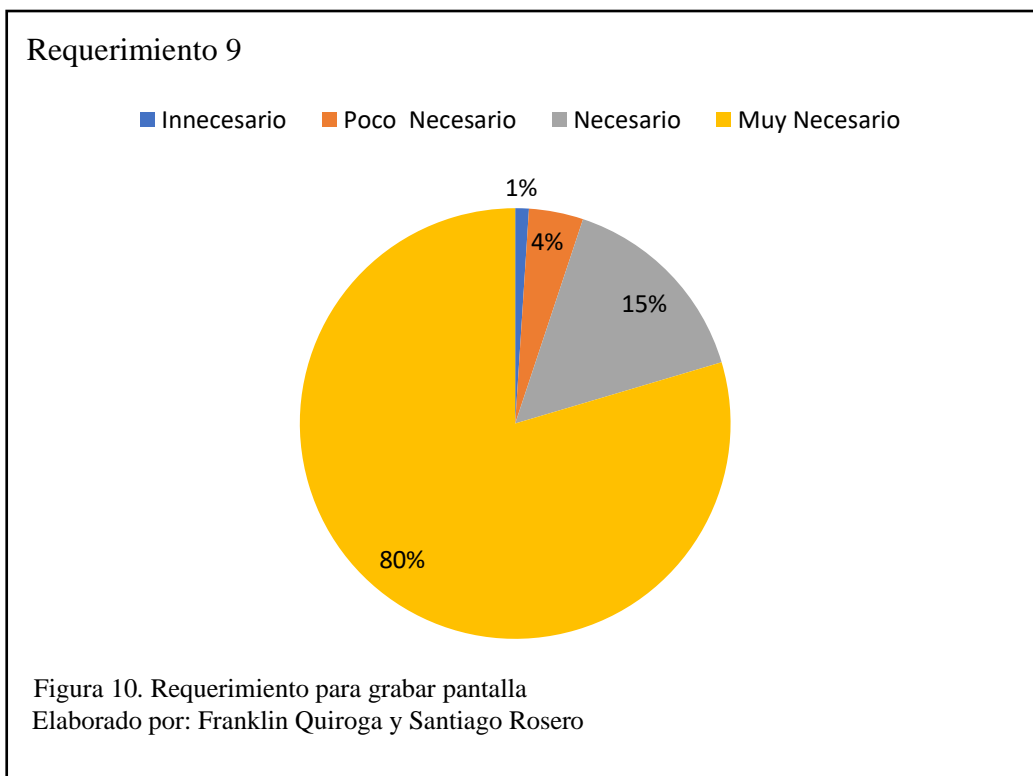
7. ¿El docente tenga la facultad de modificar y eliminar los usuarios registrados en la plataforma?



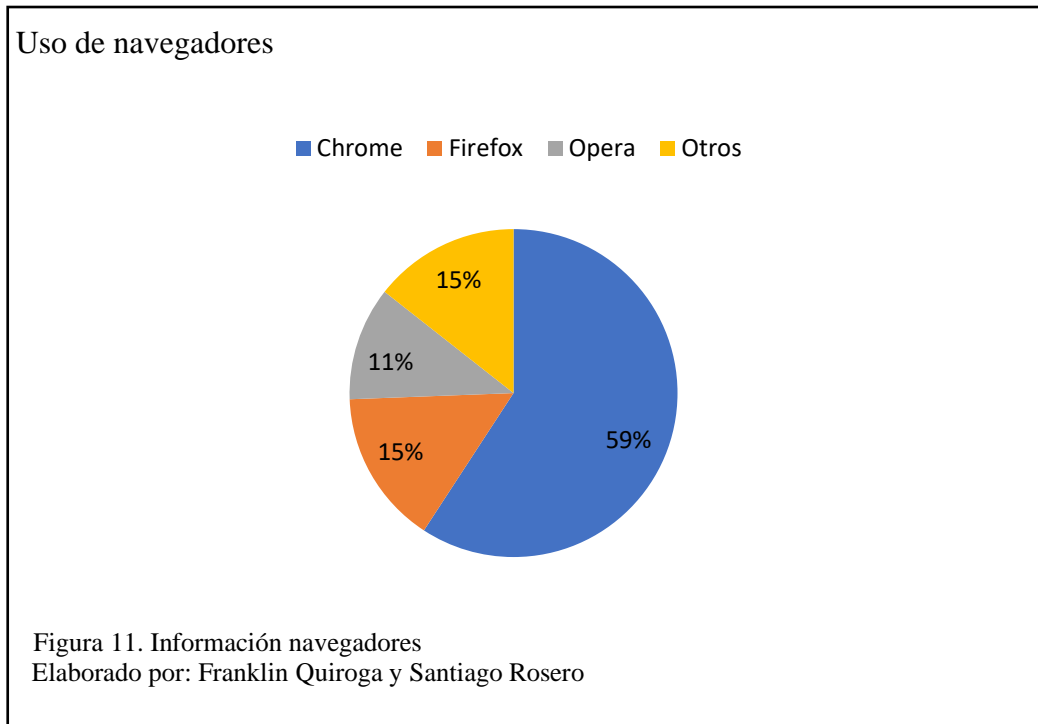
8. ¿Permitir el intercambio de archivos?



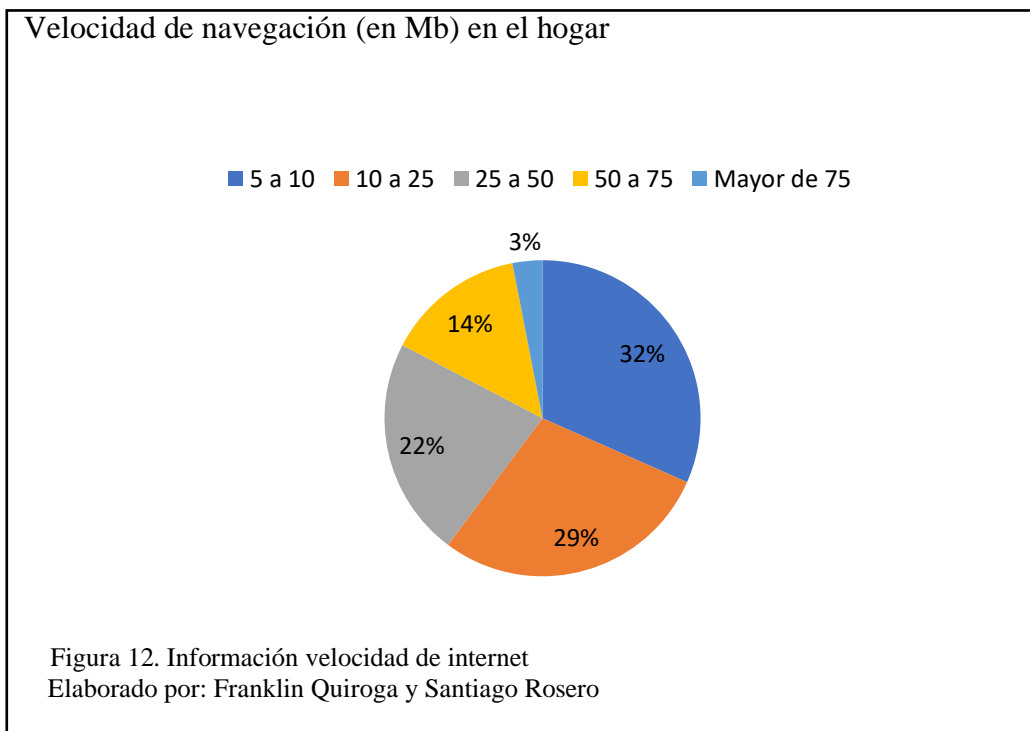
9. ¿Permitir la opción de grabar?



10. ¿Cuál navegador usa con frecuencia?



11. ¿Con qué velocidad de navegación (en Mb) cuenta en su hogar o lugar de trabajo?



De acuerdo con la información obtenida se ha decidido implementar los siguientes requerimientos funcionales y no funcionales

### 2.1.1 Requerimientos funcionales

A continuación, se detallará las reacciones que tendrá el sistema en cada uno de los módulos de acuerdo con los requerimientos solicitados por parte de los docentes y estudiantes de la carrera de Ingeniería en Sistemas y Ciencias de la Computación.

Tabla 5. Requerimientos funcionales

<b>Código</b>	<b>Requerimiento</b>
RF1	Para acceder al sistema los usuarios deberán cumplir el formulario el registro
RF2	El sistema permitirá a los usuarios registrados ingresar a la plataforma
RF3	Para ingresar al sistema de videoconferencia es necesario digitar el nombre y la respectiva contraseña proporcionada por el docente
RF4	El software permitirá crear salas, el docente tendrá la facultad de crear “n” cantidad de salas
RF5	Se podrá compartir archivos en formatos específicos
RF6	El sistema permitirá silenciar el micrófono o apagar la cámara web
RF7	El sistema permitirá una comunicación por medio de mensajes
RF8	Permitirá grabar cada clase que se realice por videoconferencia
RF9	El sistema deberá tener un botón que regrese al usuario al Home

Nota: Parámetros usados para la elaboración de la plataforma de videoconferencia

### 2.1.2 Requerimientos no funcionales

En la siguiente tabla se dará a conocer las características generales del sistema, las mismas que explican la usabilidad, seguridad y eficiencia de la aplicación de videoconferencia

Tabla 6. Requerimientos no funcionales

<b>Código</b>	<b>Requerimiento</b>
RNF1	El sistema de videoconferencia contará con manuales de usuario.

RNF2	Las contraseñas de cada usuario del sistema deben ser encriptados, con el objetivo de dar mayor seguridad con los datos de cada participante.
RNF3	El sistema brindará mensajes de error o éxito de las acciones que se realice.
RNF4	La plataforma tendrá una interfaz amigable con los usuarios.
RNF5	El sistema estará disponible solamente cuando el docente lo requiera.
RNF6	Los datos de los usuarios registrados podrán ser solamente modificados por el administrador.
RNF7	Los archivos subidos serán almacenados por un corto tiempo.
RNF8	El software de videoconferencia contará con tiempo indefinido.
RNF9	Conocer qué usuarios se encuentran activos en la plataforma.

Nota: Parámetros usados para la elaboración de la plataforma de videoconferencia.

## 2.2 Historias de usuarios

En base a las sugerencias planteadas por los usuarios, se describe las funcionalidades que tendrá el software en base a los requerimientos previamente mencionados, con el propósito de impulsar soluciones creativas para el fácil uso de la plataforma por parte de cada usuario.

- Para acceder al sistema los usuarios deberán cumplir el formulario el registro

Tabla 7. Historia de usuario Registro

Número: RF1	Usuario: Docentes y estudiantes
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos Estimados: 1	Iteración Asignada: 1
<p>Flujo normal:</p> <ol style="list-style-type: none"> <li>1. Los usuarios deberán pulsar el botón de “Register” para ingresar al módulo de registro</li> <li>2. El sistema muestra el formulario que debe de ser llenado para tener acceso al sistema</li> <li>3. Los usuarios deben ingresar la información solicitada: nombre, correo, contraseña y verificación de contraseña, el campo Rol es un dato estático.</li> <li>4. Se debe tener en cuenta que cada campo tiene su validación correspondiente.</li> <li>5. El sistema valida la información y registra al usuario en la base de datos.</li> </ol>	



6. La plataforma notificará mediante un mensaje que el usuario ha sido registrado exitosamente.
<p>Flujo alternativo:</p> <p>a) El sistema comprueba que la caja de texto “Name” no tenga números ni caracteres especiales y una longitud de máximo 25 letras, si no se cumple con los parámetros mencionados la caja de texto se tornará de color rojo, la plataforma notificará que se ajuste la información a los parámetros solicitados.</p> <p>b) El software comprobará que el correo ingresado sea válido, caso contrario la caja de texto se volverá de color rojo y notificará que se debe ingresar una dirección de correo.</p> <p>c) Si el usuario no ingresa las mismas contraseñas en el apartado de “Password” y “Confirm Password” el sistema notificará mediante un mensaje que las contraseñas no coinciden</p>
Observaciones: La información será almacenada en una base de datos y la contraseña será encriptada para que exista una mayor confidencialidad en los datos

Nota: Historia de usuario en base al requerimiento funcional 1

- El sistema permitirá a los usuarios registrados ingresar a la plataforma

Tabla 8. Historia de usuario Ingreso

Número: RF2	Usuario: Docentes, estudiantes y Administrador
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos Estimados: 1	Iteración Asignada: 1
<p>Flujo normal:</p> <ol style="list-style-type: none"> <li>1. El usuario deberá presionar el botón de “Login” para iniciar sesión.</li> <li>2. El sistema mostrará un formulario que debe ser llenado para poder acceder a la plataforma.</li> <li>3. El usuario deberá ingresar los datos: correo y contraseña con los que se registró previamente y debe seleccionar el rol al que pertenece.</li> <li>4. El sistema valida la información ingresada con la base de datos.</li> </ol>	
Flujo alternativo:	

<p>a) Si el sistema reconoce que no se ha ingresado un correo electrónico válido, la caja de texto se tornará rojo y notificará que se introduzca una dirección de correo válida.</p> <p>b) Si la contraseña o rol ingresados no son los correspondientes al correo del usuario registrado, el sistema mostrará un mensaje que la contraseña o el rol no son los correctos.</p>
<p>Observaciones: Si el usuario olvida su contraseña, este deberá comunicarse con el Administrador para que su contraseña sea Restablecida.</p>

Nota: Historia de usuario en base al requerimiento funcional 2

- Para ingresar al sistema de videoconferencia es necesario digitar el nombre y la respectiva contraseña proporcionada por el docente

Tabla 9. Historia de usuario Sistema de videoconferencia

Número: RF3	Usuario: Estudiantes
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos Estimados: 1	Iteración Asignada: + de 1
<p>Proceso:</p> <ol style="list-style-type: none"> <li>1. El usuario debe ingresar a “Profile” para acceder a la sala de videoconferencia.</li> <li>2. El usuario deberá ingresar en la caja de texto “Room” la información proporcionada por el docente y presionar el botón “Enter”.</li> <li>3. Para ingresar a la videoconferencia el usuario debe llenar el formulario con su nombre y contraseña de la sala.</li> <li>4. El sistema validará que el nombre del usuario contenga solamente letras y una longitud máxima de 12 caracteres, este no debe tener números ni caracteres especiales.</li> <li>5. Si la información es ingresada correctamente el sistema permitirá ingresar al usuario a la sala de videoconferencia.</li> </ol>	

Flujo alternativo:

- a) Si el sistema detecta que la contraseña ingresada por el usuario es diferente a la contraseña proporcionada por el docente, se enviará un mensaje indicando que la información ingresada no es la correcta.
- b) La plataforma reconocerá que las cajas de texto tengan información, en caso de que la información solicitado no exista el sistema devolverá un mensaje indicando cual es el posible error.

Observaciones: El programa creará la contraseña de la sala de manera aleatoria, el mismo que se encontrará en la url al momento de crear la sala

Nota: Historia de usuario en base al requerimiento funcional 3

- El software permitirá crear salas, el docente tendrá la facultad de crear “n” cantidad de salas

Tabla 10. Historia de usuario Creación sala

Número: RF4	Usuario: Docentes y administrador
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos Estimados: 2	Iteración Asignada: + de 1
Proceso: <ol style="list-style-type: none"><li>1. El docente debe ingresar a “Rooms” para crear una sala</li><li>2. Se deberá llenar los campos de Room Name y Your Name</li><li>3. Una vez llenado los campos se presionará el botón “Create Room”, el cual desplegará un mensaje.</li><li>4. Una vez creada la sala el software habilitará un hipervínculo que redireccionará a la sala de videoconferencia.</li></ol>	
Flujo alternativo: <ol style="list-style-type: none"><li>a) Si el docente no ingresa información en el formulario de “Create rooms” el sistema notificará que existe un error por tal motivo no será creada la sala</li><li>b) El sistema obliga al docente a llenar todos los campos del formulario para realizar la correcta creación de una sala.</li></ol>	

Observaciones: El docente tendrá que compartir la información de la sala con los estudiantes.

Nota: Historia de usuario en base al requerimiento funcional 4

- Se podrá compartir archivos en formatos específicos

Tabla 11. Historia de usuario Compartir archivos

Número: RF5	Usuario: Docentes y estudiantes
Prioridad en Negocio: Media	Riesgo en Desarrollo: Media
Puntos Estimados: 2	Iteración Asignada: + de 1
<p>Proceso:</p> <ol style="list-style-type: none"> <li>1. Para que el usuario pueda compartir información deberá dar clic en el botón “Seleccionar archivo”</li> <li>2. El sistema mostrará el Explorador de archivos el cual permitirá seleccionar el documento que se desea compartir.</li> <li>3. Para que el archivo sea subido al servidor se deberá presionar el botón “Upload”</li> <li>4. El usuario podrá subir archivos con extensiones definidas.</li> </ol>	
<p>Flujo alternativo:</p> <ol style="list-style-type: none"> <li>a) Si el usuario seleccionó el archivo, pero no presionó el botón “Upload”, el sistema no lo mostrará en la lista de archivos subidos</li> <li>b) Cuando el usuario suba un archivo, el sistema mostrará un mensaje indicando que el archivo subido se completó de manera exitosa.</li> </ol>	
Observaciones: Los archivos almacenados serán eliminados pasado cierto tiempo	

Nota: Historia de usuario en base al requerimiento funcional 5

- El sistema permitirá silenciar el micrófono o apagar la cámara web

Tabla 12. Historia de usuario Opciones

Número: RF6	Usuario: Docentes y estudiantes
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alto
Puntos Estimados: 3	Iteración Asignada: + de 1
<p>Proceso:</p> <ol style="list-style-type: none"> <li>1. Para que el usuario pueda interactuar con los demás participantes, este tiene la facultad de habilitar el micrófono y la cámara web</li> </ol>	

2. El sistema contará con los botones “Show video” y “Mute” para usar el audio y video.
Flujo alternativo: a) Si el usuario no cuenta con una cámara web o micrófono, el sistema desplegará una ventana de color anaranjado, como resultado el usuario no podrá interactuar con los demás participantes, sin embargo, el resto de las funcionalidades seguirán disponibles.
Observaciones: Los botones estarán integrado en la pantalla principal de videoconferencia.

Nota: Historia de usuario en base al requerimiento funcional 6

- El sistema permitirá una comunicación por medio de mensajes

Tabla 13. Historia de usuario Mensajes

Número: RF7	Usuario: Docentes y estudiantes
Prioridad en Negocio: Media	Riesgo en Desarrollo: Baja
Puntos Estimados: 1	Iteración Asignada: + de 1
Proceso: 1. Para que el usuario pueda enviar mensajes deberá pulsar el botón “message” 2. El sistema desplegará la caja de chat 3. El usuario debe escribir el mensaje que desea enviar a los demás participantes y aplastar el botón “Enter” para que pueda ser visualizado.	
Flujo alternativo: a) De no requerir usar esta función, el módulo permanecerá oculto.	
Observaciones: Cada mensaje incorporará el nombre del usuario y la hora de participación.	

Nota: Historia de usuario en base al requerimiento funcional 7

- Permitirá grabar cada clase que se realice por videoconferencia

Tabla 14. Historia de usuario Grabar

Número: RF8	Usuario: Docentes y estudiantes
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos Estimados: 3	Iteración Asignada: + de 1
Proceso:	

<ol style="list-style-type: none"> <li>1. Para que el usuario pueda grabar la clase deberá usar el botón “Record”</li> <li>2. El usuario podrá elegir el tipo de grabación</li> <li>3. El usuario podrá finalizar la grabación cuando él lo desee</li> </ol>
<p>Flujo alternativo:</p> <ol style="list-style-type: none"> <li>a) Si el usuario desee grabar el video de la cámara, el sistema cambiara de color el botón “Record” que permite esta acción.</li> <li>b) Cuando el usuario quiera grabar la pantalla, el botón que permite dicha acción cambiará de color a rojo, a su vez el sistema mostrará un mensaje que indica que se está compartiendo pantalla.</li> <li>c) Cada vez que el usuario desee detener la grabación, deberá dar clic en el botón, inmediatamente el sistema descargará el video de la respectiva grabación.</li> </ol>
<p>Observaciones: Se contará con un botón para finalizar la grabación de la pantalla.</p>

Nota: Historia de usuario en base al requerimiento funcional 8

- El sistema deberá tener un botón que regrese al usuario al Home

Tabla 15. Historia de usuario Botón de regreso

Número: RF9	Usuario: Docentes y estudiantes
Prioridad en Negocio: Medio	Riesgo en Desarrollo: Bajo
Puntos Estimados: 1	Iteración Asignada: + de 1
<p>Proceso:</p> <ol style="list-style-type: none"> <li>1. Si el usuario desea abandonar la videoconferencia, solo tendrá que oprimir el botón “Leave Room”</li> </ol>	
<p>Flujo alternativo:</p> <ol style="list-style-type: none"> <li>a) Si el usuario de manera accidental presiona el botón “Leave”, el sistema lo redireccionará a el módulo de “Home”.</li> </ol>	
<p>Observaciones: Esta opción es solamente una salida de la sala actual, sin embargo, el usuario seguirá dentro de la plataforma.</p>	

Nota: Historia de usuario en base al requerimiento funcional 9

## 2.3 Diagramas UML

A continuación, se muestra los diagramas UML de casos de uso, diagramas de flujo y diagramas de secuencias para los requerimientos antes mencionados.

### 2.3.1 Registro del usuario en el sistema

#### Caso de uso

Tabla 16. Especificaciones para el registro de usuario

Caso de Uso	Registro de Usuario
Actores	Funciones
Docente, Estudiante	Ingresar datos Asignar Credenciales Almacenamiento de información en la BDD

Nota: Funciones que realiza el módulo registrar usuario

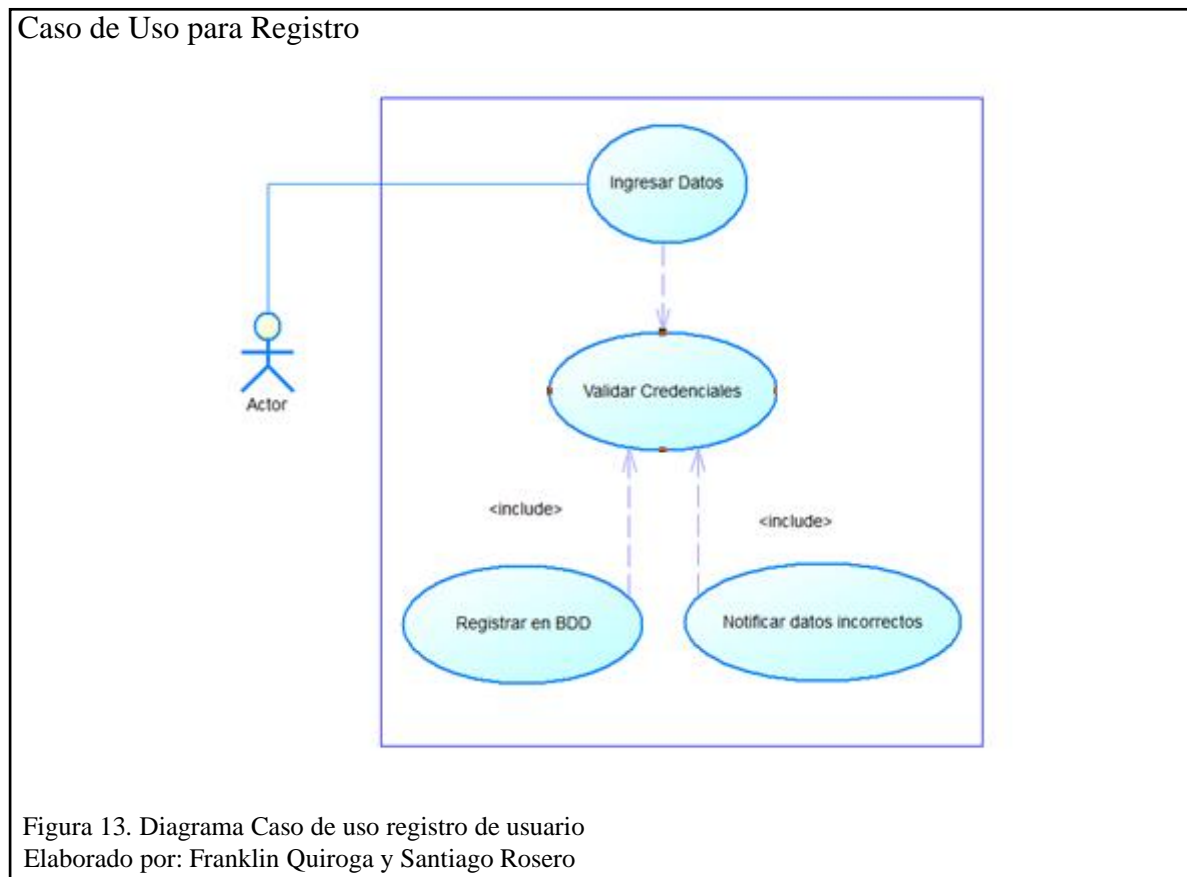


Figura 13. Diagrama Caso de uso registro de usuario  
Elaborado por: Franklin Quiroga y Santiago Rosero

## Diagrama de Flujo

Diagrama de Flujo para Registro

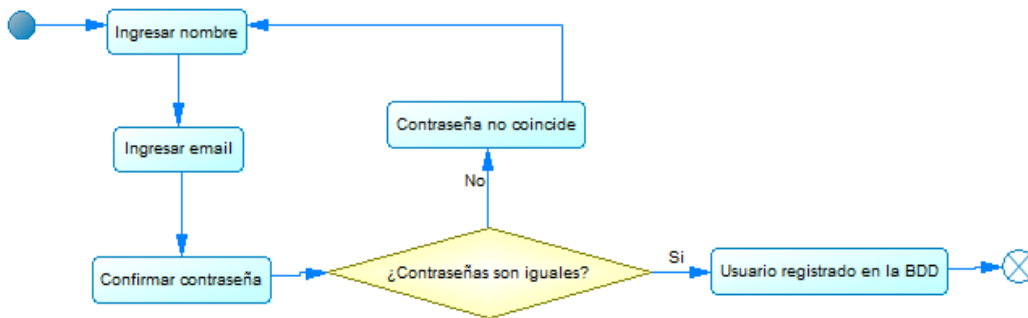


Figura 14. Diagrama de flujo para Registro de usuario  
Elaborado por: Franklin Quiroga y Santiago Rosero

## Diagrama de Secuencia

Diagrama de secuencia para registro

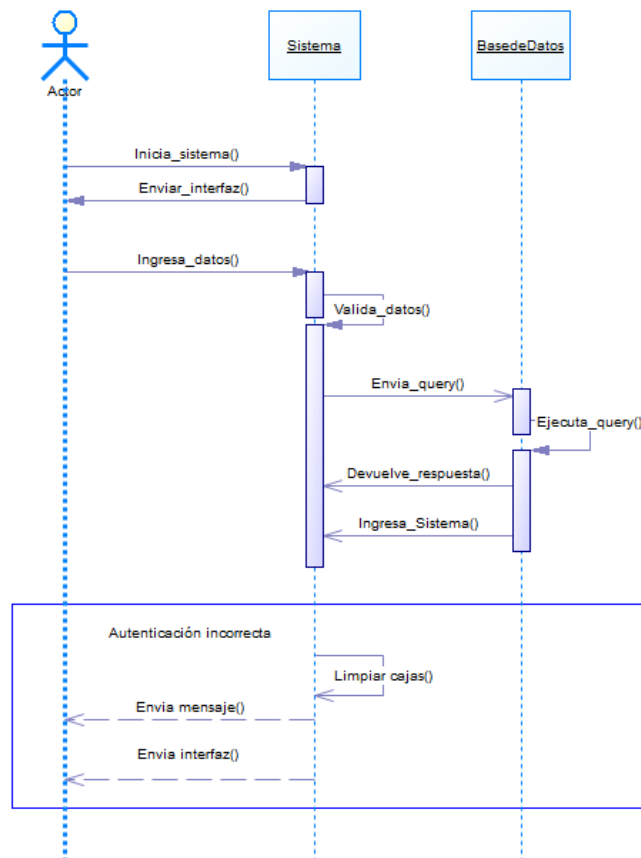


Figura 15. Diagrama de secuencia para registro de usuario  
Elaborado por: Franklin Quiroga y Santiago Rosero



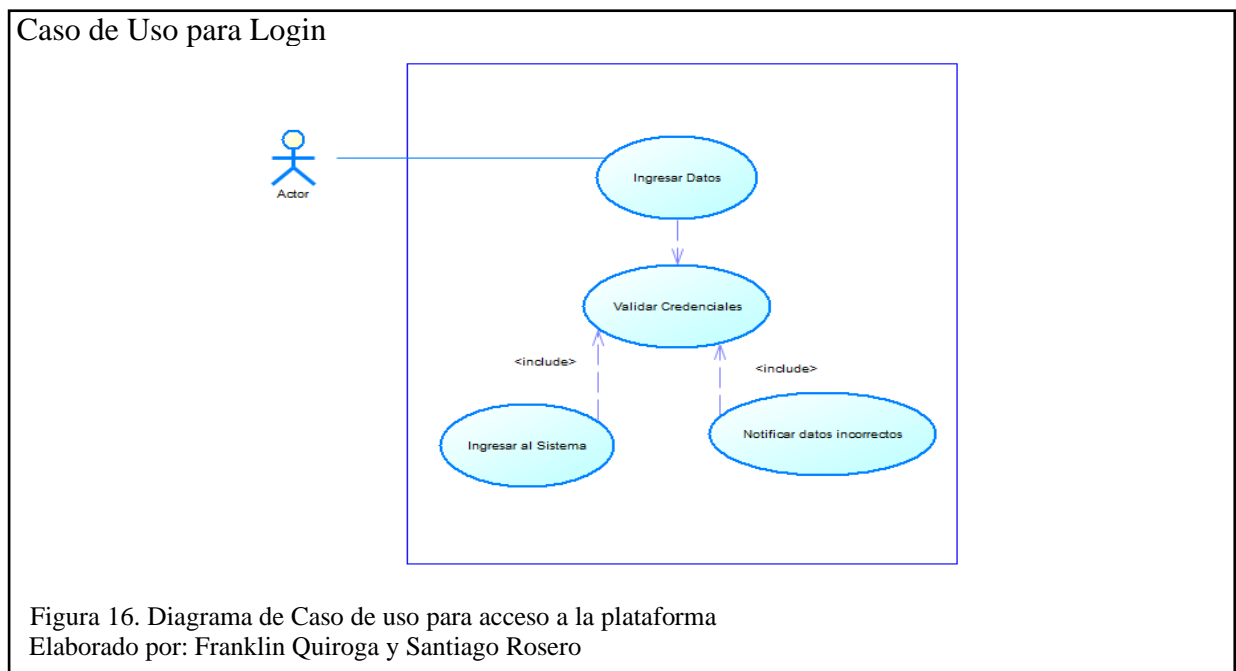
### 2.3.2 Login de usuario

#### Caso de uso

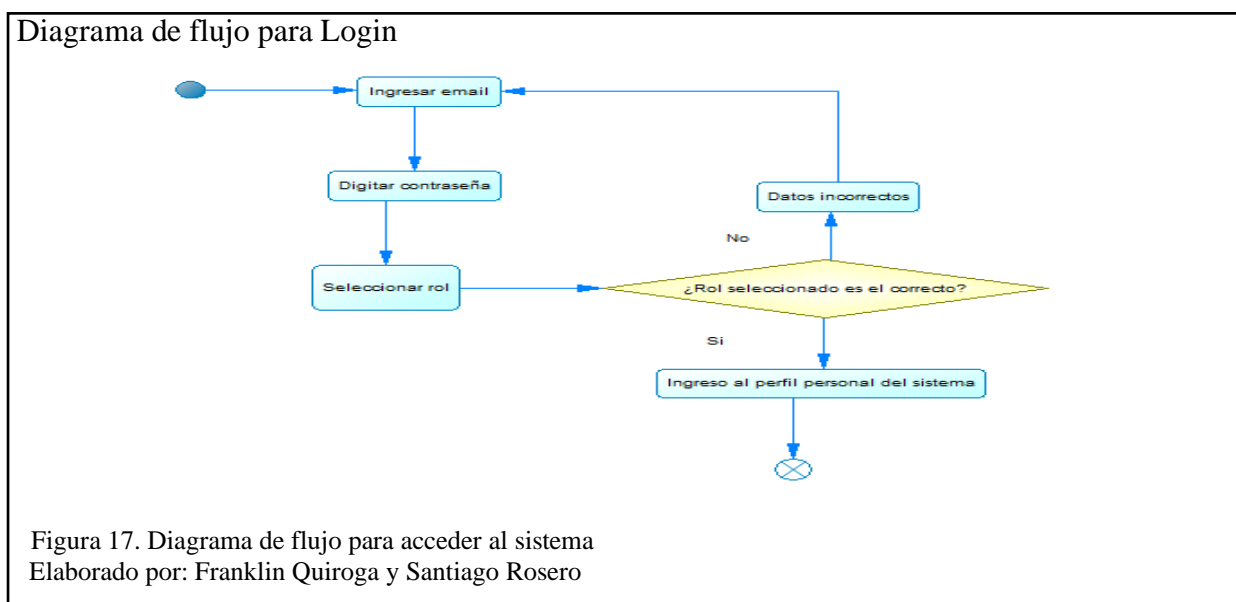
Tabla 17. Especificaciones para el logeo

Caso de Uso	Login de Usuario
Actores	Funciones
Docente, Estudiante, Administrador	Ingresar credenciales Acceder a la plataforma

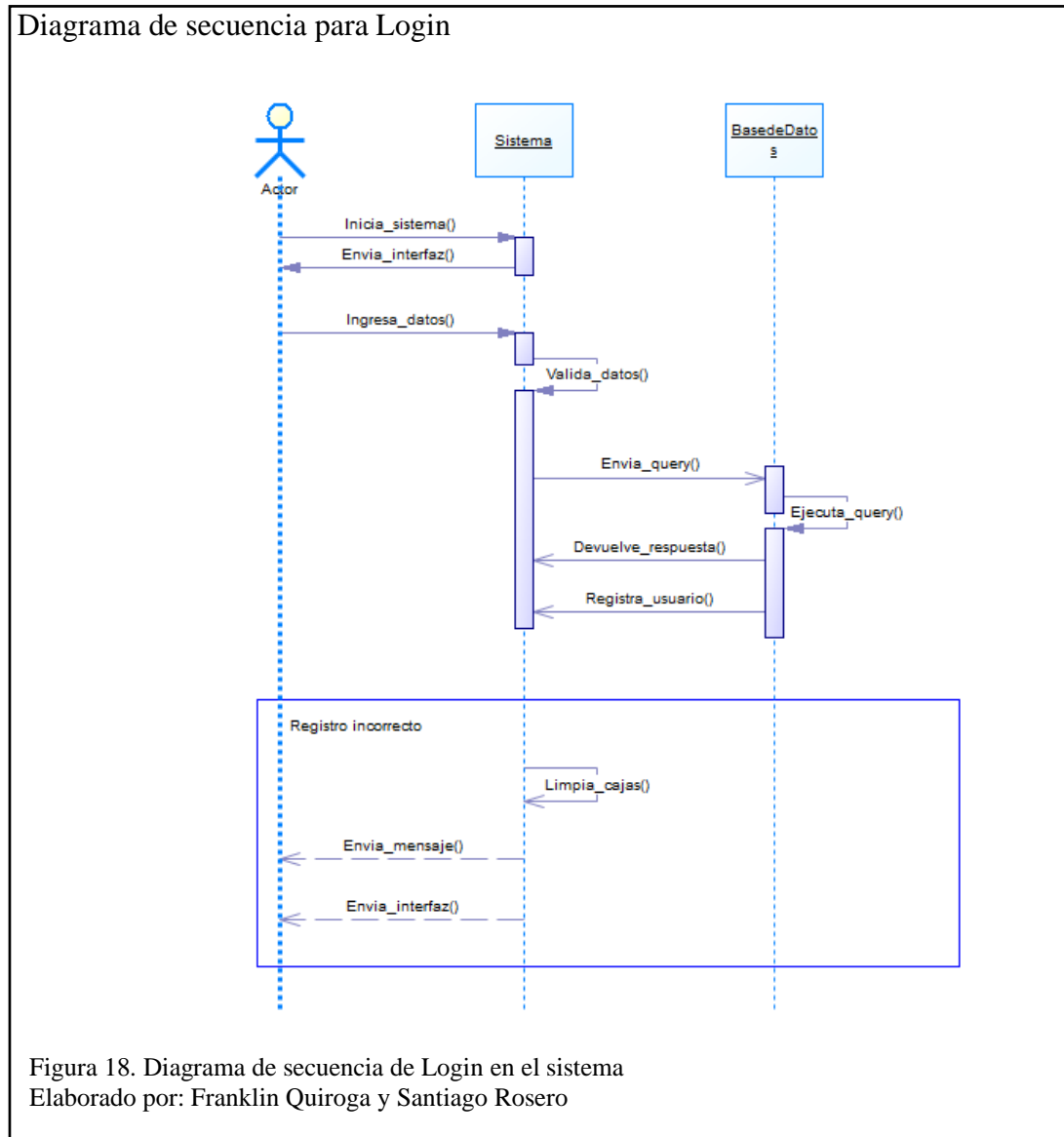
Nota: Funciones que realiza el usuario en el módulo de Login



#### Diagrama de flujo



## Diagrama de Secuencia



### 2.3.3 Creación de Sala

#### Caso de uso

Tabla 18. Especificaciones para salas

Caso de Uso	Creación de sala
Actores	Funciones
Docente	Ingresar de datos Habilitar sala

Nota: Funciones que realiza el docente en el módulo de creación de sala

### Caso de Uso para creación de sala

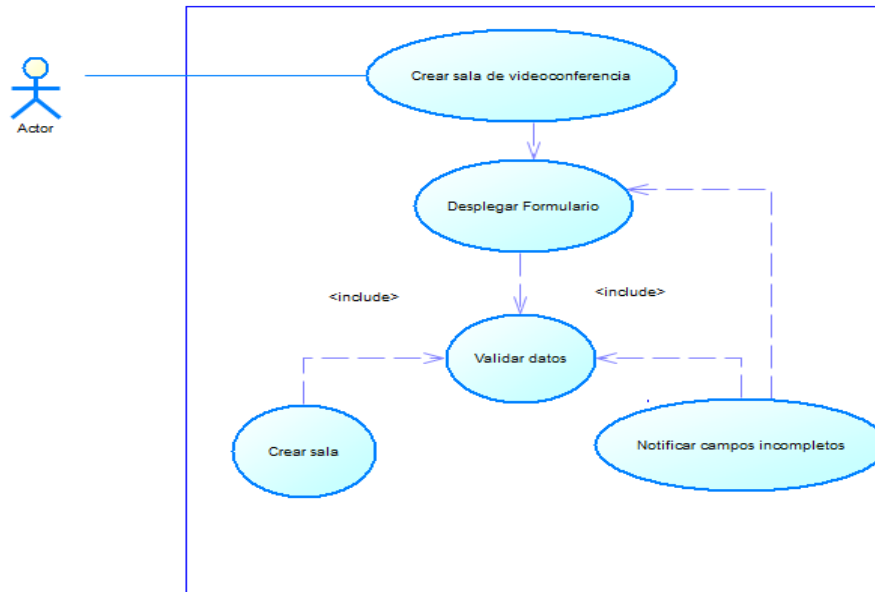


Figura 19. Diagrama Caso de uso para crear una sala de videoconferencia  
Elaborado por: Franklin Quiroga y Santiago Rosero

### Diagrama de flujo

#### Diagrama de flujo para establecer una sala

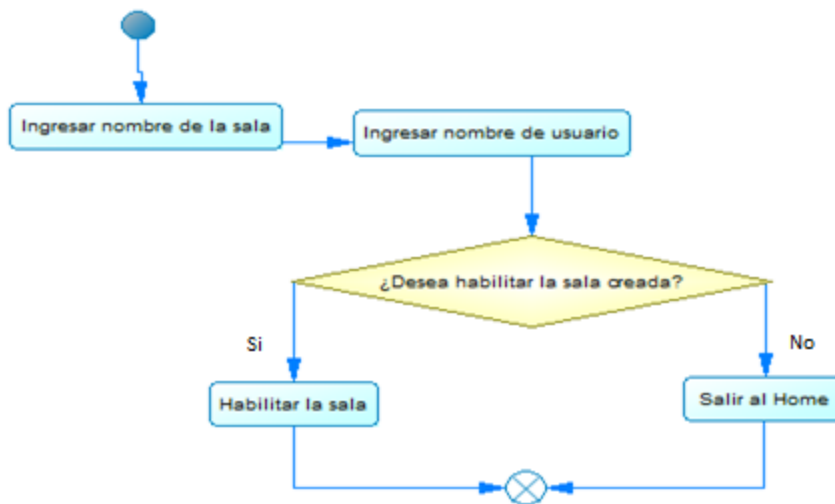
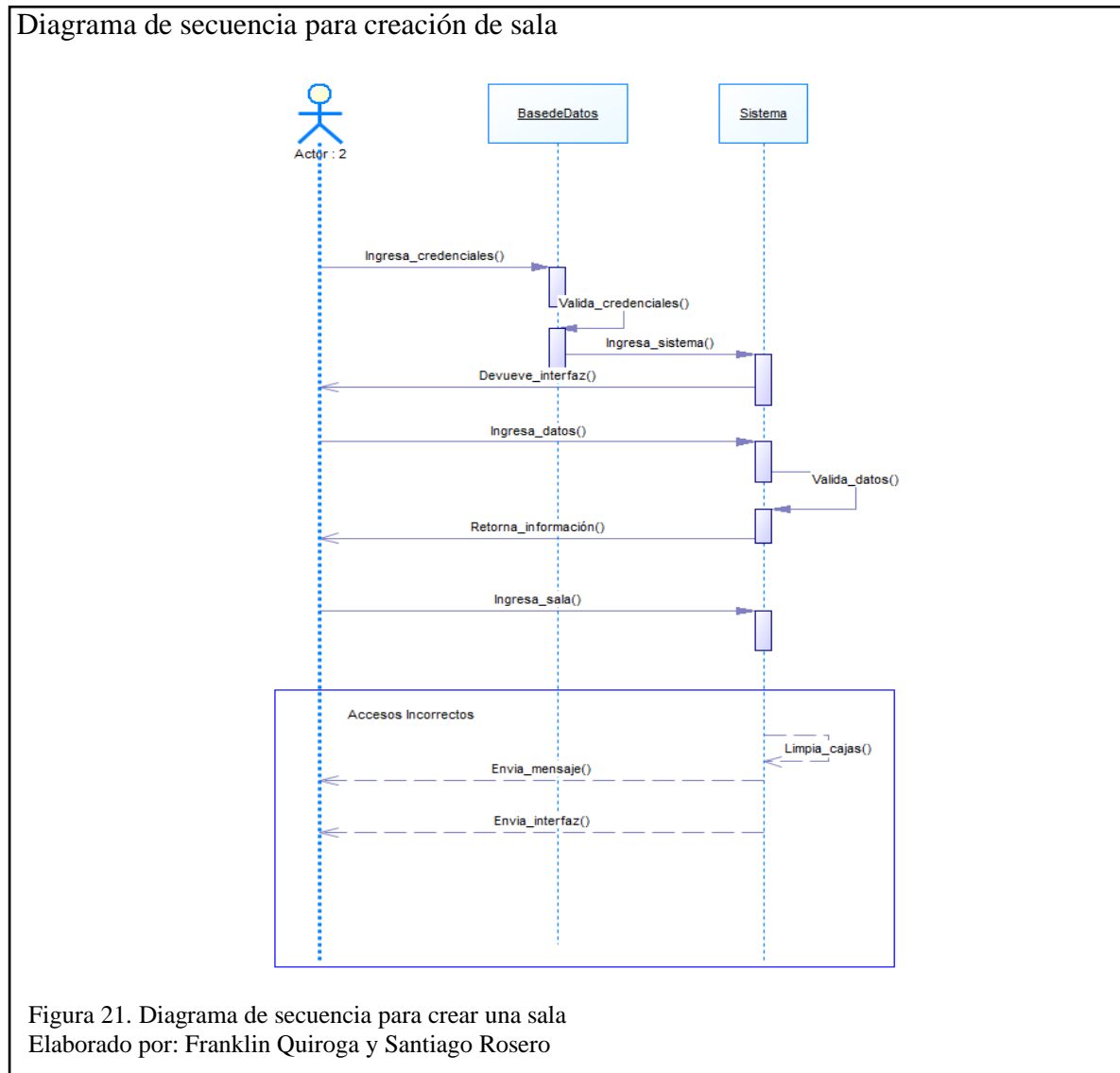


Figura 20. Diagrama de flujo para crear una sala  
Elaborado por: Franklin Quiroga y Santiago Rosero

## Diagrama de Secuencia



### 2.3.4 Ingreso a la sala

#### Caso de uso

Tabla 19. Especificaciones para ingreso a la sala

Caso de Uso	Ingreso a la sala
Actores	Funciones
Estudiante	Ingresar de datos Conexión a sala habilitada

Nota: Funciones que realiza el estudiante para acceder a la sala respectiva.

### Caso de Uso para ingreso a la sala

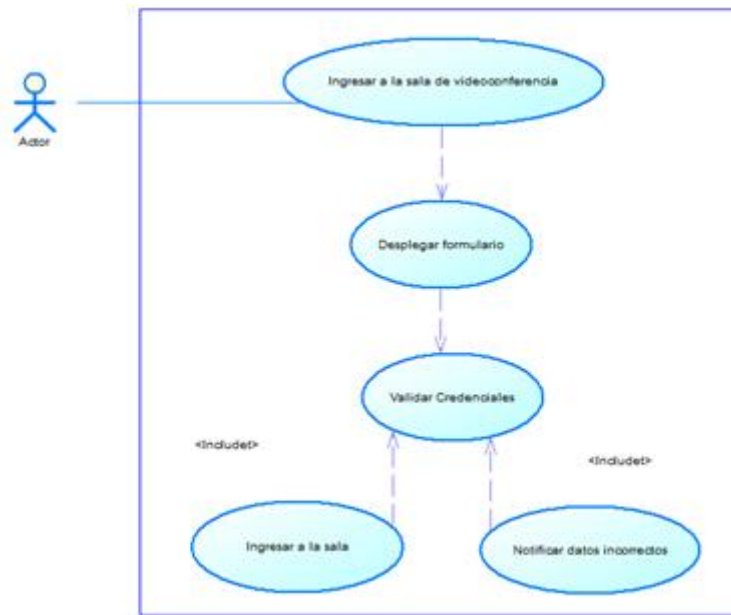


Figura 22. Diagrama Caso de uso para acceder a la sala de videoconferencia  
Elaborado por: Franklin Quiroga y Santiago Rosero

### Diagrama de flujo

#### Diagrama de flujo para ingresar a una sala por parte del estudiante

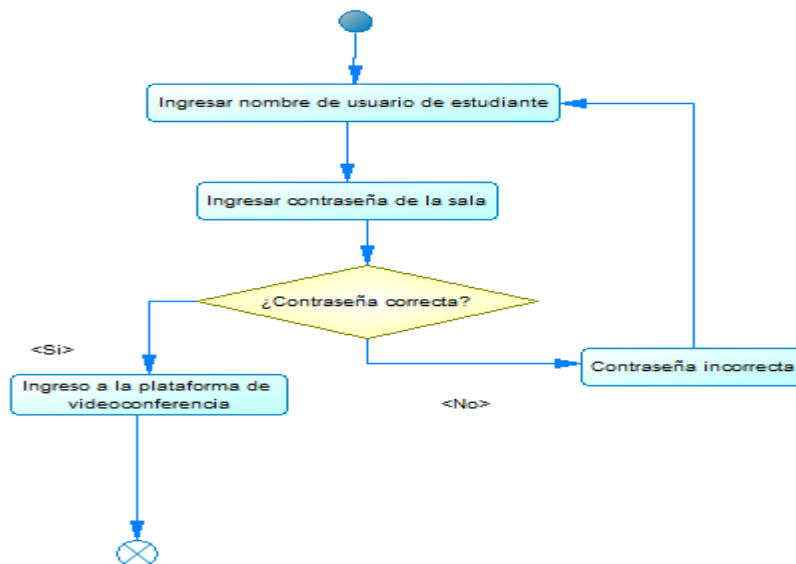
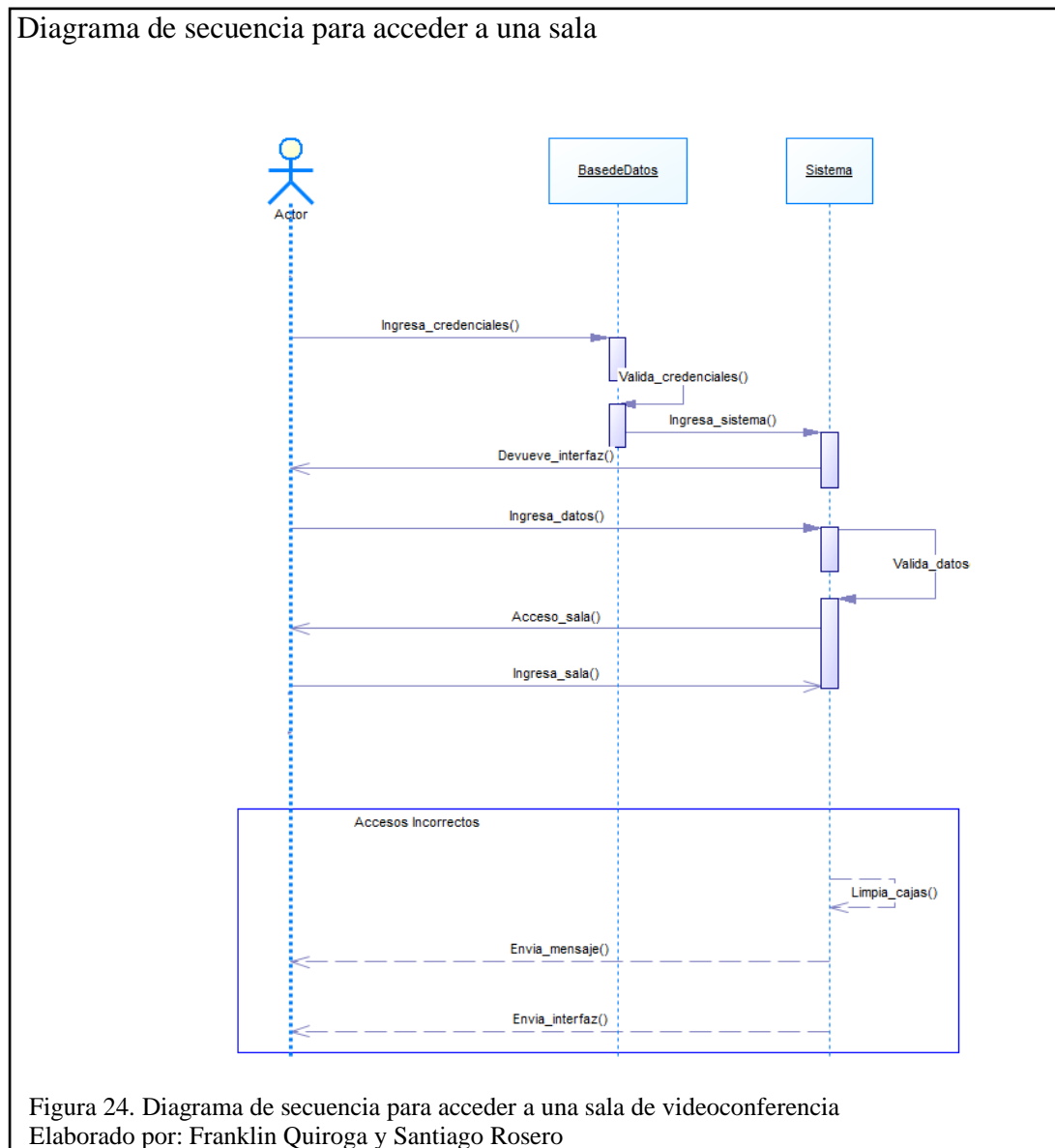


Figura 23. Diagrama de flujo para acceder a una sala  
Elaborado por: Franklin Quiroga y Santiago Rosero

## Diagrama de Secuencia



### 2.1.1 Funcionalidades de la sala de videoconferencia

#### Caso de uso

Tabla 20. Especificaciones de las funcionalidades de videoconferencia

Caso de Uso	Funcionalidades de una sala de videoconferencia
Actores	Funciones
Estudiante, Docente	Prender cámara y micrófono Enviar mensajes Compartir pantalla Grabar pantalla

	<b>Subir archivos</b> <b>Listar contenido del servidor</b>
--	---

Nota: Características de una sala de videoconferencia

### Caso de Uso de funcionalidades de una sala de videoconferencia

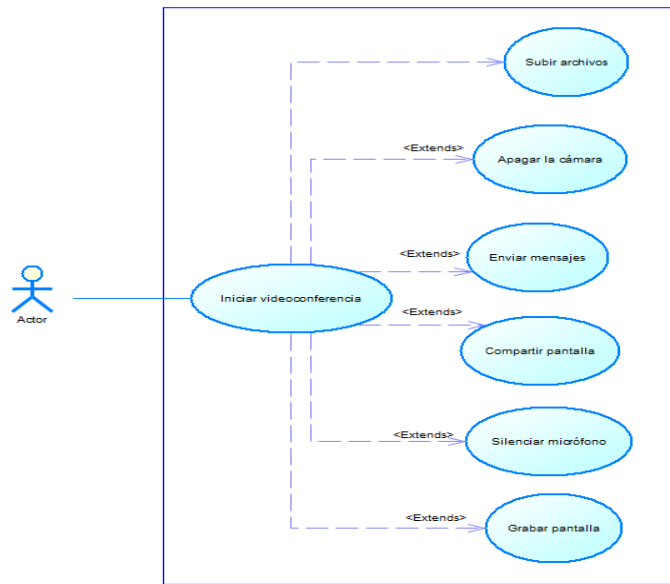
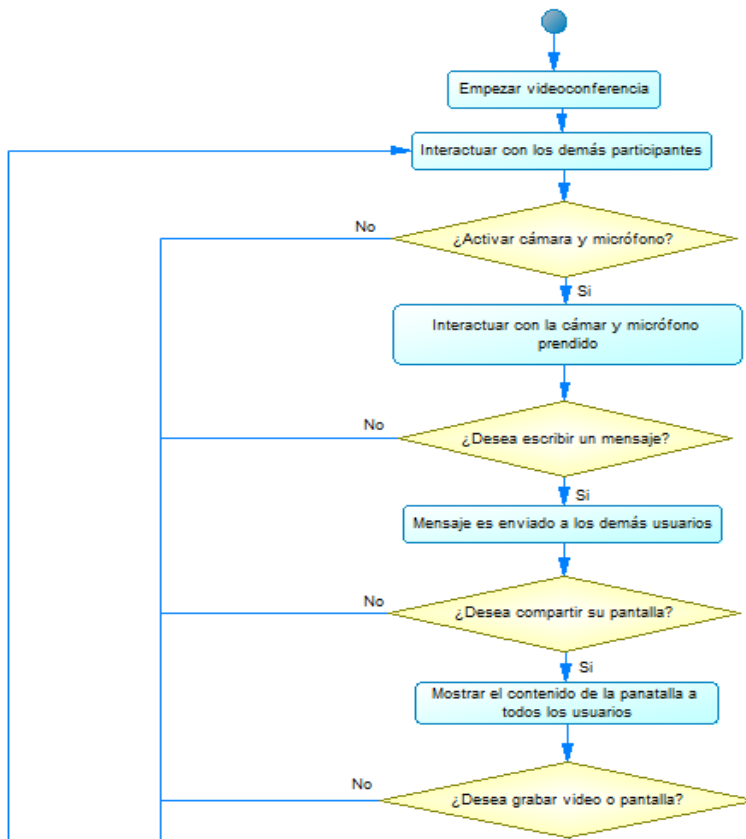


Figura 25. Diagrama Caso de uso para participar en la sala  
 Elaborado por: Franklin Quiroga y Santiago Rosero

### Diagrama de flujo

#### Diagrama de flujo para participar en una sala



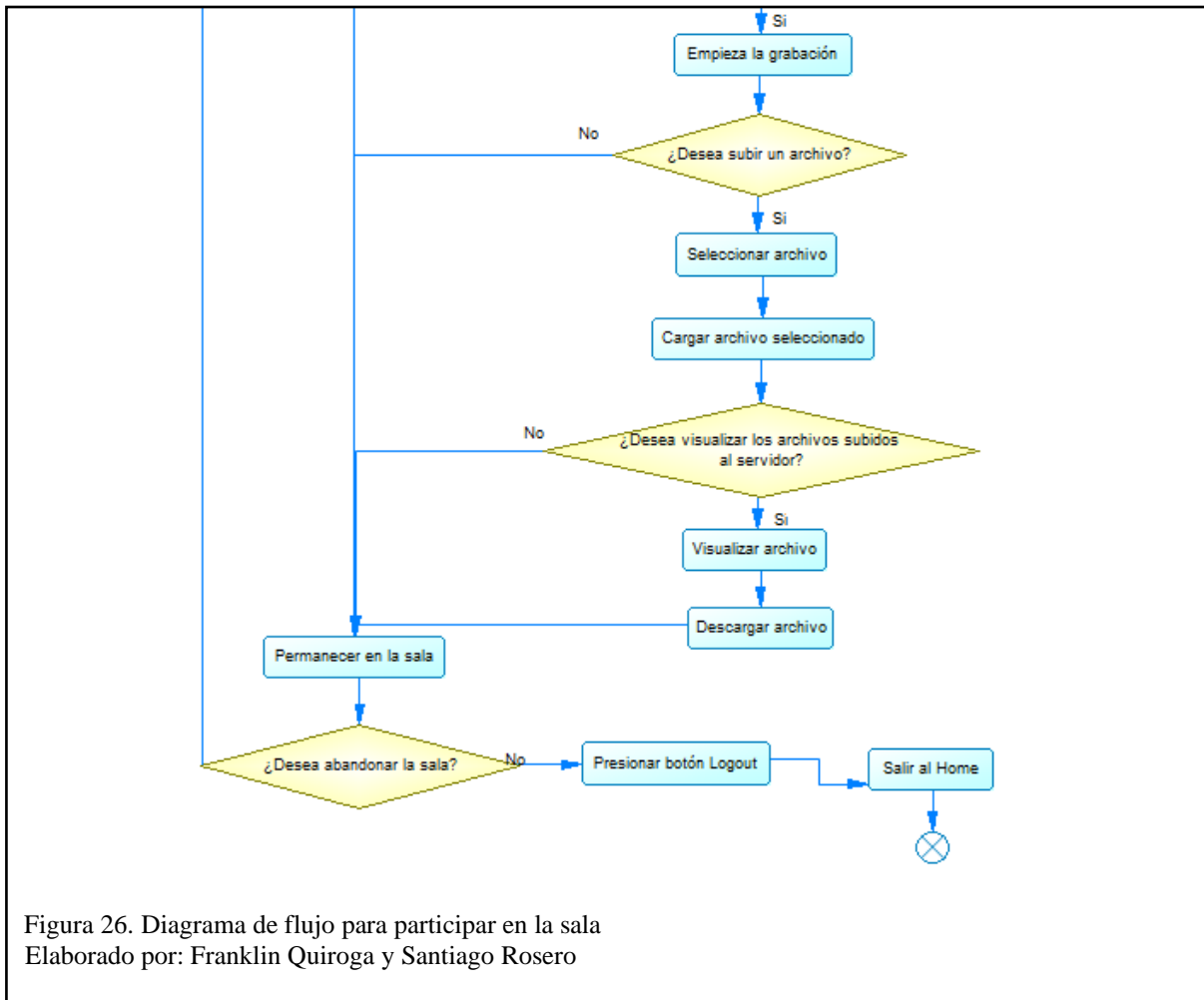
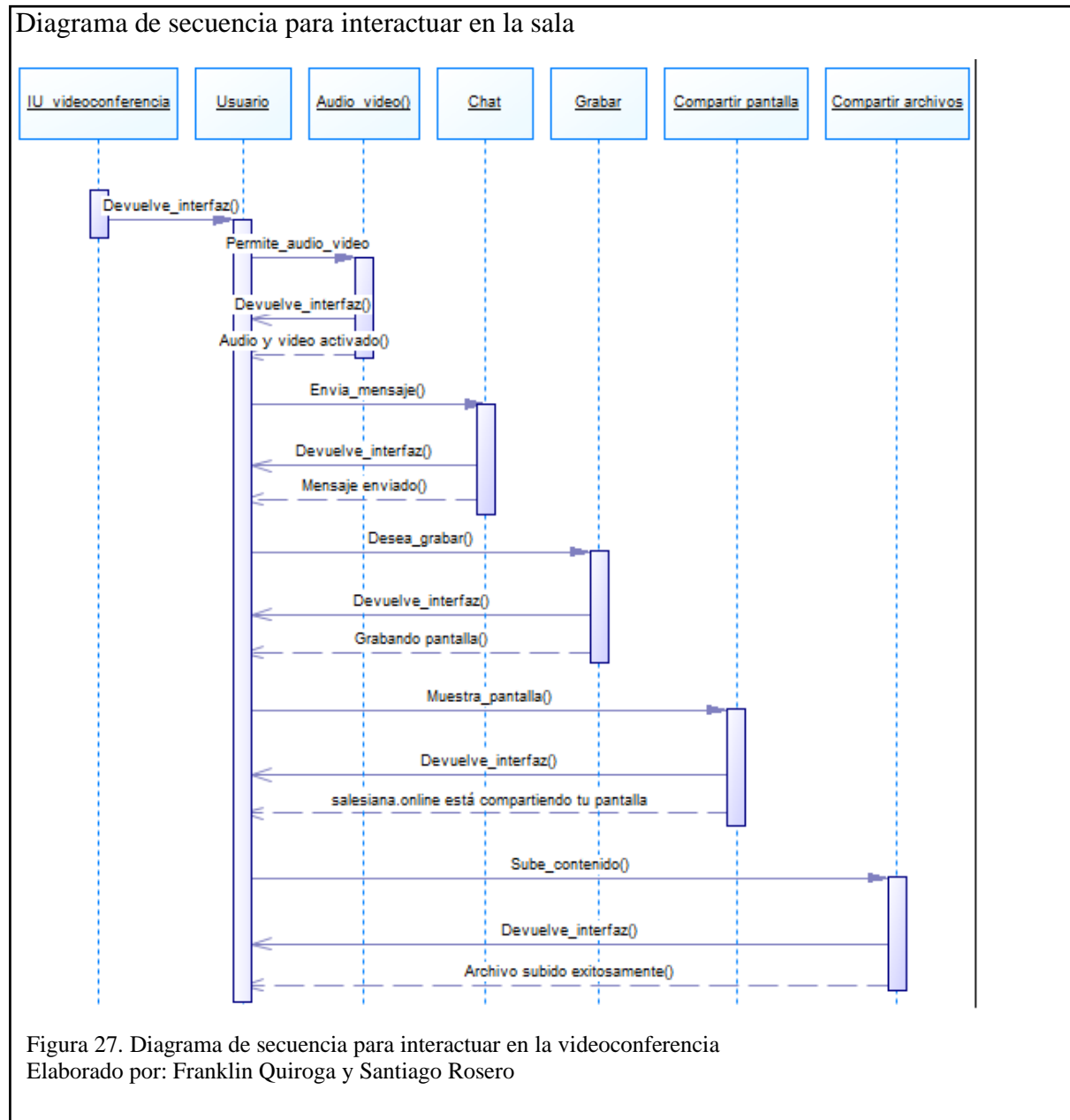


Figura 26. Diagrama de flujo para participar en la sala  
 Elaborado por: Franklin Quiroga y Santiago Rosero



## Diagrama de secuencia



## 2.4 Análisis de Factibilidad

### 2.4.1 Factibilidad Económica

#### Costo de Desarrollo

La creación del software está a cargo de estudiantes de la Universidad Politécnica Salesiana con el objetivo de poder obtener el título de Ingeniería en Sistemas. Los costos del proyecto en base a las funcionalidades de software se pueden observar en la Tabla 21. Para el cálculo se

tomó en consideración el número de horas invertidas y el costo por hora que cobraría un programador junior.

Tabla 21. Costo en base a funcionalidades

<b>Funcionalidades</b>	<b>Horas</b>	<b>Costo/hora</b>	<b>Costo a pagar</b>
Administración de usuarios.	32	5	160,00
Aplicación Híbrida.	52	5	260,00
Sitio Web	32	5	160,00
Carga de archivos.	20	5	100,00
Certificados de seguridad.	4	5	20,00
Cifrado de datos.	52	5	260,00
Integración con aplicaciones de terceros.	20	5	100,00
Integración con audio y video.	52	5	260,00
Integración de chat	12	5	60,00
Interfaz de usuario personalizada.	32	5	160,00
Número de usuarios en tiempo real	4	5	20,00
Perfiles de usuario.	20	5	100,00
Sincronización a través de distintos dispositivos.	52	5	260,00
Diseño de base de datos	52	5	260,00
<b>Total</b>	<b>436</b>		<b>2180,00</b>

Nota: Costo a pagar en base a horas invertidas en el desarrollo

### **Costo de Hardware**

Para el desarrollo del proyecto se contrató el servicio de Google Cloud, con el fin de tener una misma solución: creación de instancias de máquinas virtuales y administración de redes VPC.

El uso de un dominio se vio necesario para que los usuarios puedan acceder a la plataforma de una forma más eficiente.

Los costos de dominio, cloud y servidor turn para la plataforma de videoconferencia se detallan a continuación:

Tabla 22. Costos de Hardware

<b>Servicio</b>	<b>Nombre del Servicio</b>	<b>Precio(anual)</b>
Dominio	Namecheap(salesiana.online)	\$ 24.73
Cloud	Google Cloud (16GB)	\$ 300
Servidor turn	Xirsys	\$995
Total		\$ 1319.73

Nota: Valores en la adquisición de hardware para la plataforma de videoconferencia

El dominio “salesiana.online” tuvo un costo de \$24, sin embargo, se debe tener en cuenta que este pago se lo realiza anualmente.

Google Cloud tuvo un costo de \$300, que corresponde al servicio de cloud, el cual brinda los siguientes beneficios como son: monitorizar máquinas virtuales, consultar los registros de VM configurar reglas de cortafuego, por dichas razones se ha decidido utilizar Google cloud para crear una instancia de MV en Linux que permita personalizar el entorno que se adapte a las configuraciones requeridas en el desarrollo del software. Dichas configuraciones no son admisibles en un hosting.

Para la transmisión de datos entre los diferentes dispositivos que se conectan a la videoconferencia se ha utilizado un servidor turn, el mismo que tuvo un costo de \$995 anuales, obteniendo una cantidad específica de ancho de banda al mes.

### **Costo de Software**

En el desarrollo de la plataforma de videoconferencia se está utilizando como sistema operativo Ubuntu, el cual permite utilizar software libre, por consiguiente no se ha tenido que pagar ningún costo de licenciamiento en la adquisición de software.

## 2.4.2 Factibilidad Técnica

Para que la plataforma tenga fluidez se utilizó un servidor Ubuntu con las siguientes características como se muestran a continuación en la Tabla 23, la plataforma necesita una retroalimentación continua entre el usuario y el servidor por lo cual se utilizaron lenguajes tanto para el front end como para el back end. Se analizó y comparó varias clouds con el fin de conocer la que mejor se acople a las necesidades en la creación del software de videoconferencia.

### Hardware

Tabla 23. Hardware máquina virtual

<b>Componente</b>	<b>Especificación</b>
Tipo de máquina	n1-standard-1
Procesador	64 bits
Memoria RAM	4 Gb
Disco Duro	10 Gb (Persistente Standar)
Puertos	4

Nota: Hardware requerido para la plataforma de videoconferencia

### Software

Tabla 24. Software máquina virtual

<b>Componente</b>	<b>Especificación</b>
Sistema Operativo	Ubuntu 18.04
Navegador web	Firefox, Google Chrome, Safari
IDE	Atom, Visual Studio Code
Servidor FTP	FileZilla
Conexión SSH	MobaXterm

Nota: Software requerido en la máquina virtual para la plataforma de videoconferencia

Para la siguiente comparativa se han tomado en cuenta las cloud más conocidas y potentes a nivel mundial como lo es Google Cloud y Amazon Web Service, ya que cada una de ellas tienen

características compatibles con un proyecto escalable que requiere de máquinas eficientes como lo es una plataforma de videoconferencia.

Tabla 25. Comparativa cloud

<b>Características</b>	<b>Google Cloud</b>	<b>Amazon Web Service</b>
Tipo de Máquina	n1-standard-1 – n1-standard-96	m3.medium – m3.2xlarge m4.large – m4.16xlarge m5.large – m5d.24xlarge
CPU alta	n1-highcpu-2 – n1-highcpu-96	c4.large – c4.8xlarge c5.large – c5d.18xlarge
Precio de los discos (por GB/mes)	\$0.040 (disco estándar)	\$0.045
Precios de las multiregiones en caliente (por GB/mes)	\$0.0260 Incluye el traslado	S3 Estándar(x2) \$0.0460 Transferencia \$0.0100
Niveles	Niveles de servicio de la red	N/A
Locación GCE (us-central1-c)	Alta: 77ms, Media: 85ms, Baja: 381ms	Alta: 333ms, Media: 343.5ms, Baja: 382ms

Nota: Cloud requerido para la plataforma de videoconferencia

Nota: Recuperado de (AMAZON, 2021)

En base a la evidencia presentada en la tabla 25, los parámetros más significativos para escoger la solución de Google Cloud es que cuenta con el tipo de máquina que más se adapta a las necesidades de la plataforma junto con la CPU. Los precios son más accesibles y su tiempo de respuesta es más rápida en base a la localización de la máquina virtual dentro del servidor de Google Cloud.

### **2.4.3 Factibilidad operativa**

La plataforma de Web RTC permite realizar videoconferencias en tiempo real, cuenta con inicio de sesión para gestionar los datos de cada usuario, tiene una interfaz amigable con el usuario y fácil de utilizar. Además, posee seguridad y confidencialidad al momento de requerir información. El sistema se adapta a la mayoría de los navegadores utilizados en la actualidad.

Para poder tener acceso a la plataforma el usuario debe haberse registrado mediante el ingreso de datos que solicita el formulario “Register”, el sistema de registro permitirá validar que el campo “contraseña” y “confirmar contraseña” sean iguales, también se validará que el campo email contenga un correo permitido.

Al ingresar al sistema se tendrá un menú que según el perfil del usuario habilitará ciertos módulos de la plataforma de videoconferencia:

- Módulo Home: Tendrá una breve descripción del software y contará con manuales que ayudaran al estudiantes y docente a conocer más la herramienta
- Módulo Profile: Se tendrá los datos del usuario, un hipervínculo que dirige a la página oficial de la Universidad Politécnica Salesiana y un botón que permitirá ingresar a la sala de videoconferencia.
- Módulo Room: Este módulo solo es visible para el docente y el administrador, aquí el docente ingresará el nombre de la sala y el nombre con el que se visualizará en el sistema.

Al completar los datos aparecerá un hipervínculo que permitirá el acceso a la sala de videoconferencia al docente, aquí también el sistema brindará el password asignado a la sala que se acaba de crear.

- Módulo de Videoconferencia: Al ingresar al módulo de videoconferencia el usuario debe aceptar los permisos al navegador para que haga el uso de la cámara y el micrófono, también se tiene los botones de: apagar cámara, silenciar micrófono, compartir pantalla, grabar pantalla, ingresar al chat, compartir archivos, listar los archivos compartidos y salir de la sala para volver al “Home”

#### **2.4.4 Análisis costo beneficio**

Para el estudio de viabilidad del proyecto se analizaron tres diferentes puntos de vista, con el objetivo de tener las mejores herramientas que se adapten al desarrollo de la plataforma. De esta manera el sistema tendrá un funcionamiento eficiente y factible que se alinea al presupuesto establecido.

Se analizó la factibilidad económica, en donde se contempló el costo de desarrollo, el valor a pagar sería de \$2180, sin embargo, dicho valor no es tomado en consideración en el análisis ya que los responsables del desarrollo fueron estudiantes que tienen como objetivo obtener el título de Ingeniería en Sistemas.

El análisis del costo en hardware y software arrojó un resultado positivo, para iniciar con el estudio de factibilidad técnica, al contar con una instancia de máquina virtual dentro de una cloud con características específicas se pudo comprobar que al ejecutar el software este no presenta ningún problema de rendimiento y el sistema trabaja con normalidad, al tener un dominio es más fácil acceder a la plataforma web, de igual manera se configuró un certificado SSL para mejorar la seguridad, se ha visto la necesidad de utilizar un servidor turn que permita retransmitir el tráfico entre las conexiones de los usuarios.

Por último, se realizó una investigación de la factibilidad operativa a cerca de los módulos necesarios que debe tener la plataforma para que cumpla con las expectativas del usuario.

En base al trabajo realizado se identificó que para el usuario es importante contar con una interfaz gráfica amigable, de igual manera es imprescindible que tenga documentación que explique el funcionamiento del software de manera clara y sencilla. Por tal motivo se afirma que la factibilidad para realizar el proyecto es positiva, ya que cumple con todos los requerimientos que los usuarios han solicitado.

Después de haber estudiado los diferentes análisis se procedió a realizar una comparativa entre otras soluciones que brindan el servicio de videoconferencia, para lo cual se determinó lo siguiente.

Tabla 26. Comparativa entre soluciones de videoconferencia

		<b>Licenciamiento por un administrador</b>		<b>Licenciamiento por 20 administradores</b>
<b>Herramienta</b>		<b>Mensual</b>	<b>Anual</b>	<b>Anual</b>
<b>Otras soluciones</b>	<b>Zoom</b>	\$ 49,00	\$588	\$11760
	<b>Microsoft Teams</b>	\$20	\$240	\$4800
<b>Solución propia de la universidad</b>	<b>Incluye valor de desarrollo</b>	\$20,83	\$249,98	\$4999,61
	<b>No incluye valor de desarrollo</b>	\$7,86	\$94,27	\$1885,33

Nota: Comparativa de costo de licenciamiento entre soluciones de videoconferencia

Como se puede observar en la Tabla 26 el contar con un software propio no resulta ser costoso a diferencia de otras soluciones, ya que en el actual proyecto además de proporcionar las funciones comunes de videoconferencia, se tiene la opción de adicionar funcionalidades que se desee implementar en el sistema propio.



## CAPITULO III

### CONSTRUCCIÓN PLATAFORMA WEB DE VIDEOCONFERENCIA

#### 3.1 Base de datos

Para el presente proyecto se ha utilizado una base de datos relacional como lo es MySQL, se ha visto que es la más viable debido a que: es de código abierto, tiene un gran rendimiento, resulta ser eficiente al realizar búsquedas de datos, es compatible con diferentes sistemas operativos como Linux y Windows y requiere de pocos recursos de CPU y memoria al momento de usarla.

Se escogió esta base de datos ya que se está trabajando solamente con una tabla, la cual contiene los datos de cada usuario que se registra en el sistema, el mayor beneficio de usar MySQL es la rapidez al momento de hacer una consulta a la base de datos.

##### 3.1.1 Esquemas para la base de datos

A continuación, se presentará la estructura de la tabla “users”:

- La base de datos contará con 6 campos, en los cuales se almacenará la información de los usuarios que se hayan registrado en la plataforma.
- Los campos “name”, “email” y “rol” serán de tipo varchar con longitud de 100 caracteres.
- Contará con un campo auto incrementable que en este caso será “id” y tendrá una longitud de 11 caracteres.
- Los campos “password” y “confirm\_pass” deberán ser de tipo varchar con longitud de 255, ya que la contraseña es encriptada para controlar su seguridad.

Tabla 27. Campos de la base de datos

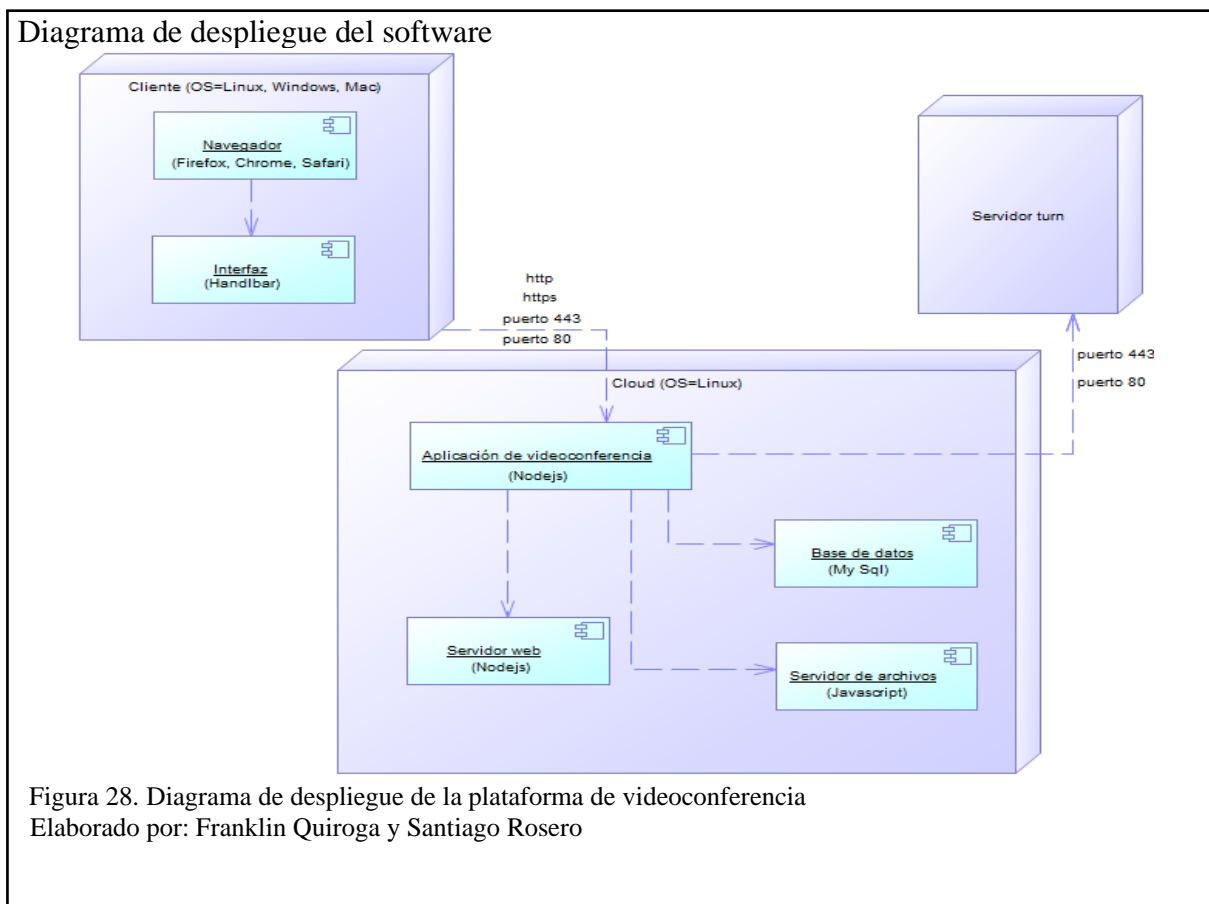
Nombre	Tipo	Extra
id	Int(11)	AUTO_INCREMENT
name	Varchar(100)	N/A
email	Varchar(100)	N/A
password	Varchar(255)	N/A
confirm_pass	Varchar(255)	N/A
rol	Varchar(100)	N/A

Nota: Campos de la base de datos que utiliza la sala de videoconferencia

## 3.2 Desarrollo e Implementación

### 3.2.1 Diagrama de Despliegue

A continuación, se visualizará el diagrama de despliegue, el cual mostrará las relaciones físicas del sistema, las cuales detallan el funcionamiento de la plataforma.



Cliente: Es el usuario que desea conectarse con la aplicación web mediante el uso de un navegador con su respectiva interfaz, a su vez se utilizarán los puertos y protocolos 443, 80, http y https para su navegación y comunicación con la cloud.

Cloud: Infraestructura en la cual se encontrará la base de datos que contiene la información de cada usuario, la aplicación de videoconferencia que se encuentra montada en una máquina virtual que está alojada dentro de la GCP, el servidor web el mismo que permite procesar la aplicación web, por último, el servidor de archivos que contiene los archivos que enviará al cliente como respuesta de su petición.

Servidor turn: Su función principal es retransmitir el tráfico en cuanto a peticiones que requiere la videoconferencia entre los participantes. Tendrá una comunicación directa con la cloud a través del puerto 443 y 80.

### **3.2.2 Diagrama de navegación**

A continuación, se mostrará el flujo de interacción que usuario tiene con los diferentes módulos con los que cuenta la plataforma de videoconferencia.

Diagrama de navegación del software

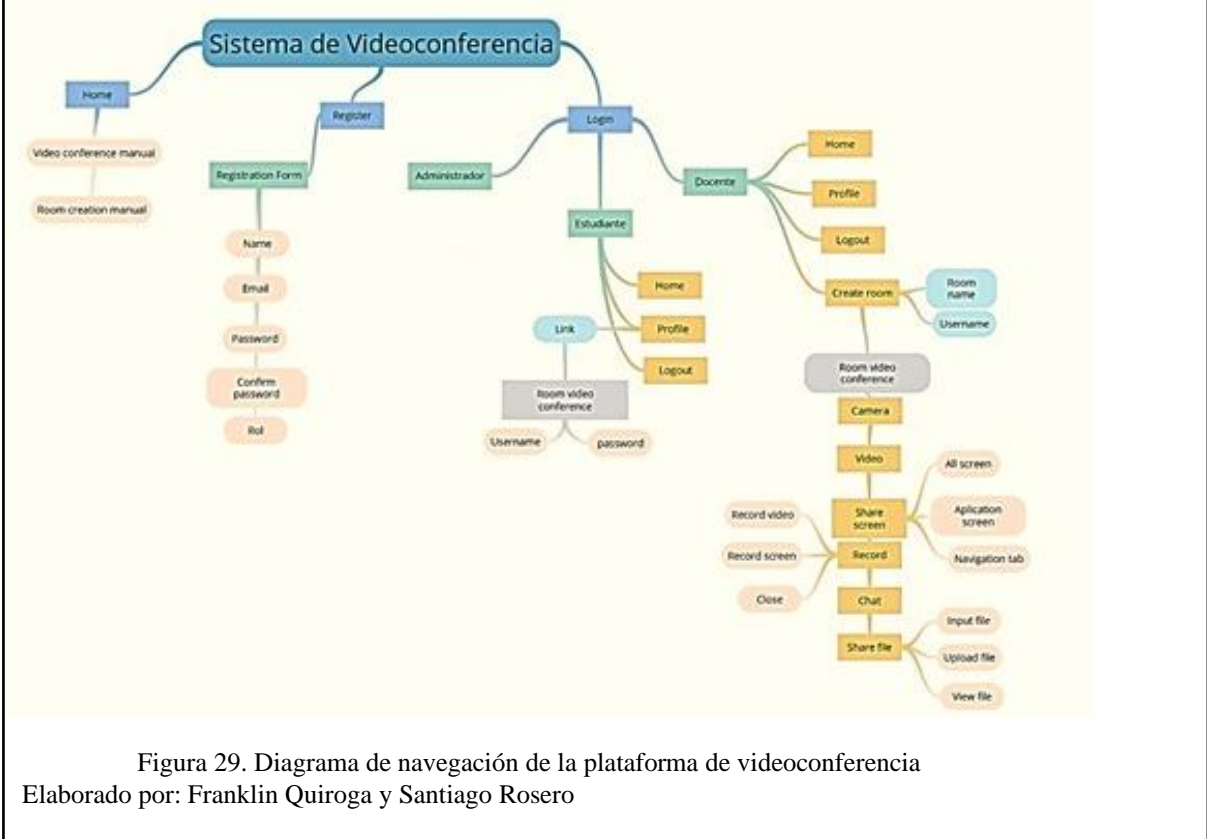


Figura 29. Diagrama de navegación de la plataforma de videoconferencia  
Elaborado por: Franklin Quiroga y Santiago Rosero

### 3.3 Componentes del sitio web

A continuación, se explicará cómo se encuentra estructurado del proyecto.

Tabla 28. Estructura de la plataforma de videoconferencia

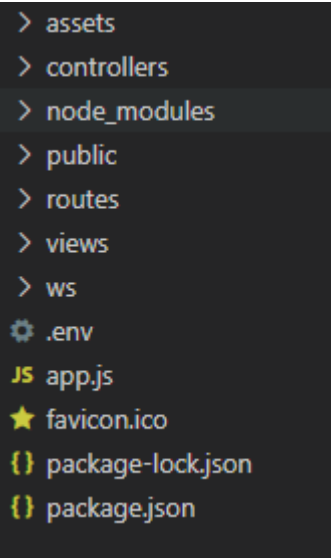
CARPETA	ARCHIVOS	DESCRIPCIÓN
/assets/js	autolink.js, events.js, helpers.js, rtc.js	Estos archivos cuentan con toda la programación necesaria para la videoconferencia y las funcionalidades descritas en los requerimientos funcionales.
/assets/upload		Dentro del directorio “upload” se almacenarán los archivos que se han subido desde la plataforma web
/controllers	auth.js	Aquí se tiene la codificación para controlar la autenticación y registro de los usuarios mediante el

		uso de sentencias SQL hacia la base de datos.
/public	style.css	Este archivo permite personalizar el aspecto del sitio web con el objetivo de mejorar la interacción entre el usuario y la plataforma.
/routes	auth.js, pages.js	Dentro de estos archivos se puede encontrar el código de las diferentes rutas que tiene el proyecto como son el login y register.
/views	index_doc.hbs, index_est.hbs, index.hbs, login.hbs, profile_est.hbs, profile.hbs, register.hbs, room.hbs	Contiene los archivos handlebar, los mismo que tiene el código para la interfaz gráfica para el usuario.
/ws	stream.js	Este archivo contiene la configuración de los sockets, los mismos que son necesarios para la comunicación entre el cliente y servidor.
/	.env	Dentro de este archivo se encuentra los diferentes parámetros que son utilizados para la conexión con la base de datos.
/	app.js	Este archivo contiene la configuración necesaria para que el servidor pueda ser ejecutado y este a su vez pueda interactuar con el resto de carpetas.
/	package.json	Dentro de este archivo se puede encontrar las dependencias y

		versiones con las que se ha trabajado en el proyecto.
--	--	---

Nota: Directorios y archivos del sistema de videoconferencia

Estructura de los componentes de sistema de videoconferencia



```

> assets
> controllers
> node_modules
> public
> routes
> views
> ws
⚙ .env
JS app.js
★ favicon.ico
{} package-lock.json
{} package.json

```

Figura 30. Estructura de componentes  
Elaborado por: Franklin Quiroga y Santiago Rosero

### 3.4 Código importante

En este apartado se explicará segmentos de códigos más relevantes de cada módulo con el objetivo de detallar el funcionamiento de la plataforma, las cuales son: creación del servidor web, conexión a la base de datos con sus respectivas consultas, estructura del proyecto

Para el desarrollo de la aplicación web se ha utilizado el framework Express que trabaja como una estructura cliente servidor. Una de las ventajas al utilizar Express es que tiene una buena correlación con Nodejs ya que permite crear aplicaciones de buen rendimiento y que sean escalables, lo que resulta una característica importante en el desarrollo del sitio web. Al trabajar por módulos la integración de los distintos escenarios no resultó complicado.

### 3.4.1 Módulo de Inicio, Registro y Acceso a la plataforma

#### 3.4.1.1 Creación del servidor

Como primer paso para la elaboración del sitio web es necesario instalar Express como una dependencia, definir el puerto por el cual se va a entablar una comunicación y crear la carpeta “routes” que contendrá las diferentes ventanas de navegación.

Configuración puerto de escucha
<pre>const express = require("express"); const app = express(); app.use('/', require('./routes/pages')); app.listen(443, () =&gt; {   console.log("Server started on port 443"); })</pre>
Figura 31 Servidor web Elaborado por: Franklin Quiroga y Santiago Rosero

Como se observa en la Figura 31 primero se requiere inicializar express en la variable “app”, mediante dicha variable se hace el llamado a las diferentes rutas de navegación y el método “app.listen” permite definir el puerto por el cual va a inicializar la comunicación.

#### 3.4.1.2 Conexión base de datos

Debido a que se manejarán los módulos Login y Register se optó por utilizar como base de datos “MySQL”, el mismo que permitirá almacenar la información de los formularios ya antes mencionados, para poder entablar la comunicación es necesario instalar mysql como una dependencia. Como se visualiza en la Figura 32 es necesario definir los siguientes parámetros: host, user, password, database, estos datos se obtendrán al momento de crear la base de datos.

#### Configuración de parámetros de base de datos

```
const mysql = require ('mysql');
const db = mysql.createConnection({
  host: Nombre_de_la_máquina,
  user: Nombre_de_usuario_mysql,
  password: Contraseña_de_mysql,
  database: Nombre_de_la_base_de_datos
});
```

Figura 32. Conexión a base de datos  
Elaborado por: Franklin Quiroga y Santiago Rosero

#### 3.4.1.3 Interfaz de usuario

Para que el usuario pueda visualizar las diferentes ventanas es necesario inicializar “path”. Mediante el método “path.join” el cual puede especificar rutas donde se encuentran los archivos necesarios para mejorar la usabilidad del sitio web.

En cada interfaz se ha optado por implementar Bootstrap para lograr una interfaz intuitiva y amigable con el usuario.

Se ha utilizado handlebars como plantilla el mismo que permitirán crear interfaces para la página web. Como se observa en la Figura 33 el sitio web solo podrá ejecutar archivos con extensión \*.hbs

#### Ruta de los archivos handlebar

```
const path = require('path');
const publicDirectory = path.join(__dirname, 'public')
app.use(express.static(publicDirectory));
app.set('view engine', 'hbs');
```

Figura 33. Ruta archivos handlebars  
Elaborado por: Franklin Quiroga y Santiago Rosero

#### 3.4.1.4 Login y Register

Para las ventanas de Login y Register se ha utilizado el método POST, el mismo que permite enviar los datos de los atributos globales “id” desde el formulario hacia el archivo auth.js de JavaScript. En este archivo se encuentran las funciones: login, register y logout, las mismas que permitirán ejecutar varias acciones acordes a las necesidades del usuario.



Una vez obtenido los datos del formulario a través del método POST estos se almacenarán en el argumento de solicitud “req” como se puede evidenciar en la Figura 34. De igual manera se puede observar en la misma Figura el proceso de validación que el sistema lo realiza mediante una sentencia sql el cual verifica si el correo y la contraseña coinciden, caso contrario se envía una alerta de que los datos son incorrectos, si los datos son ingresados correctamente el usuario será redireccionado mediante “redirect” hacia el home de la aplicación web

```
Query base de datos y validación
b.query('SELECT * from users WHERE email = ? ' , [email], async(error, results) =>
{
  const rol = results[0].rol;
  const email1 = results[0].email;
  const passwordd = results[0].password;

  if (rol == rol && email == email1 && password == passwordd) {
    const id = results[0].id;
    switch (rol) {
      case 'Estudiante':
        const token = jwt.sign({id}, process.env.JWT_SECRET, {
          expiresIn: process.env.JWT_EXPIRE_IN
        });

        const cookieOptions = {
          expires: new Date(
            Date.now()+ process.env.JWT_COOKIE_EXPIRES * 24 * 60 * 60 * 1000
          ),
          httpOnly: true
        }
        res.cookie('jwt', token, cookieOptions);
        res.status(200).redirect("/index_est");

        break;
      case 'Docente':
        const token1 = jwt.sign({id}, process.env.JWT_SECRET, {
          expiresIn: process.env.JWT_EXPIRE_IN
        });

        const cookieOptions1 = {
          expires: new Date(
            Date.now()+ process.env.JWT_COOKIE_EXPIRES * 24 * 60 * 60 * 1000
          ),
          httpOnly: true
        }
    }
  }
}
```

```

    res.cookie('jwt', token1, cookieOptions1);
    res.status(200).redirect("/");
    break;
    default:
  }
} else {
  res.status(401).render('login',{
    message: 'Email or password is incorrect'
  })
}
}}

```

Figura 34. Validación Login

Elaborado por: Franklin Quiroga y Santiago Rosero

De igual manera en la función Register los datos serán almacenados en el argumento de solicitud “req” y por medio de sentencias sql se podrá conocer si el correo ya se encuentra registrado la plataforma lo notificará y el usuario deberá utilizar otro correo para poder completar el formulario de registro, como se puede evidenciar en la Figura 35, este cuenta con otro proceso de validación donde las contraseñas ingresadas deben de coincidir. Una vez que el usuario haya ingresado todos los datos de manera correcta en el formulario, la plataforma por medio de la sentencia sql Insert almacenará los datos ingresados en la tabla users de la base de datos.

#### Query para registrar un usuario en la base de datos

```

const {name, email, password, passwordconfirm, rol} = req.body;
db.query('SELECT email from users WHERE email = ?', [email], async(error, results) => {
  if (error){
    console.log(error);
  }
  if(results.length > 0){
    return res.render('register', {
      message:'That email is already in use'
    })
  }else if(password !== passwordconfirm){
    return res.render('register', {
      message:'Password de not match'
    });
  }
}

let hashedPassword = await bcrypt.hash(password, 8);

```

```

console.log(hash>Password);

db.query('INSERT into users set ?', {name: name, email: email, password:password , passwordEncrypt:hashedPassword, rol: rol}, () =>{
  if(error){
    console.log(error);
  }else{
    console.log(results);
    return res.render('register', {
      message: 'User registered'
    });
  }
})
});

```

Figura 35. Validación Register  
Elaborado por: Franklin Quiroga y Santiago Rosero

Cuando el usuario se encuentra dentro de la plataforma, se tendrá la opción de salir. Como se muestra en la Figura 36 el usuario es redireccionado hacia Home de salida.

Botón para salir de la plataforma

```

exports.logout = async (req, res) => {
  res.cookie('jwt', 'logout', {
    expires: new Date(Date.now() + 2*1000),
    httpOnly: true
  });
  res.status(200).redirect('/'); }

```

Figura 36. Redireccionamiento  
Elaborado por: Franklin Quiroga y Santiago Rosero

### 3.4.2 Módulo de Videoconferencia

El módulo de videoconferencia cuenta con características relevantes como por ejemplo la función que permite crear URL automáticamente, el cual es indispensable para el docente, ya que por medio de dicho enlace se puede acceder a las diferentes salas que el docente puede crear. Una vez dentro de la plataforma de videoconferencia el usuario puede interactuar con los demás participantes por medio de las siguientes características: desactivar cámara, silenciar micrófono, compartir pantalla, grabar pantalla, subir archivos, intercambiar mensajes.

#### 3.4.2.1 Creación de Sala

En el momento que el docente desee crear una sala, este deberá ingresar la siguiente información: nombre de la sala y nombre de usuario, en el formulario respectivamente. Esta

información es necesaria para generar el URL y acceder a la plataforma como se visualiza en la Figura 37. Para generar el enlace se utiliza la herramienta Autolink de JavaScript como se observa en la Figura 38, la cual permite agregar texto y generar un hipervínculo que redireccione a la sala de videoconferencia.

<p>Información para crear una sala con el perfil de docente</p> <pre>let roomName = document.querySelector( '#room-name' ).value; let yourName = document.querySelector( '#your-name' ).value; let roomLink = `https://salesiana.online/index?room=\${ roomName.trim().replace( ' ', '_' ) } \${ helpers.generateRandomString() }`;</pre>
<p>Figura 37. Creación de sala Elaborado por: Franklin Quiroga y Santiago Rosero</p>

<p>Función para generar la URL de la sala</p> <pre>autoLink = function () {   var callback, k, linkAttributes, option, options, pattern, v;   options = 1 &lt;= arguments.length ? slice.call( arguments, 0 ) : [];   pattern = /^(^ [\s\n] &lt;[A-Za-z]*\V/?&gt;)((?:https? ftp):\\\/[\-A-Z0-9+\\u0026\\u2019@#\\\/%?=(~_ !:,.;]*[\-A-Z0-9+\\u0026@#\\\/%~()_])/gi;   if ( !( options.length &gt; 0 ) ) {     return this.replace( pattern, "\$1&lt;a href='\$2'&gt;\$2&lt;/a&gt;" );   } }</pre>
<p>Figura 38. Generar URL de la sala Elaborado por: Franklin Quiroga y Santiago Rosero</p>

### 3.4.2.2 Acceder a la sala

Una vez el docente comparta la información para acceder al sitio web el estudiante deberá conocer la contraseña, la misma que cuenta con un proceso de validación como se evidencia en la Figura 39.

Las variables “name” y “password” son los datos que ingresa el usuario en el archivo. hbs. La variable “roomLink” permite traer la url y mediante el método Split separará una parte del enlace el cual ayudará a validar la información que ingresa el estudiante con la que genera la plataforma, dicha validación le permitirá acceder al sistema o ser direccionado a la misma página.

<p>Datos para ingresar a la sala con el perfil de estudiante</p> <pre>let name = document.querySelector( '#username' ).value; let password = document.querySelector( '#password' ).value;  let roomLink =location.href;</pre>
---

```

let password1= roomLink.split ("_");
if ( name && password ) {
  if (password == password1[1]) {
    document.querySelector( '#err-msg-username' ).innerHTML = "";
    sessionStorage.setItem( 'username', name );
    location.reload();
  }else {
    send.render('/')
  }
}
}

```

Figura 39. Ingresar a la videoconferencia  
Elaborado por: Franklin Quiroga y Santiago Rosero

### 3.4.2.3 Recursos de la videoconferencia

En la estructura del sistema de videoconferencia se tiene los módulos que permiten activar las funcionalidades de la página web como se puede observar en la Figura 40.

Módulos necesarios para el funcionamiento de la videoconferencia

```

<script type="module" src='assets/js/rtc.js'></script>
<script type="module" src='assets/js/events.js'></script>

```

Figura 40. Módulo  
Elaborado por: Franklin Quiroga y Santiago Rosero

El método “getUserMedia” solicita al usuario que conceda los permisos de audio como se muestra en la Figura 41, video y compartir pantalla, si el usuario no concede dichos permisos los recursos multimedia no podrán ser utilizados por el usuario. Se asigna el valor de “true” para poder usar la cámara y micrófono del ordenador.

Método para activar el video y audio

```

getUserFullMedia() {
  if ( this.userMediaAvailable() ) {
    return navigator.mediaDevices.getUserMedia( {
      video: true,
      audio: {
        echoCancellation: true,
        noiseSuppression: true
      }
    } );
  }
  else {
    throw new Error( 'User media not available' );
  }
}

```

```
    },  
  }  
},
```

Figura 41. Estructura getUserMedia  
Elaborado por: Franklin Quiroga y Santiago Rosero

El método “getDisplayMedia” permite al usuario capturar el contenido de una pantalla o una ventana de navegación. De la misma manera se puede grabar la transmisión mediante el método MediaStream o se puede transmitir a través de una sesión de WebRTC. Como se evidencia en la Figura 42 se asigna el valor de “always” para que el video permanezca activado y el valor de “true” para poder compartir el audio en ese instante.

Método para compartir pantalla dentro de la videoconferencia

```
shareScreen() {  
  if ( this.userMediaAvailable() ) {  
    return navigator.mediaDevices.getDisplayMedia( {  
      video: {  
        cursor: "always"  
      },  
      audio: {  
        echoCancellation: true,  
        noiseSuppression: true,  
        sampleRate: 44100  
      }  
    } );  
  }  
  else {  
    throw new Error( 'User media not available' );  
  }  
},
```

Figura 42. Estructura shareScreen  
Elaborado por: Franklin Quiroga y Santiago Rosero

El método “saveRecordedStream” que se detalla en la Figura 43 permite grabar lo que proyecta la cámara web y la pantalla compartida de la plataforma de videoconferencia, una vez finalizada la grabación se descargará un archivo con la extensión “webm”, que a su vez utiliza el codec de video VP9. Este archivo podrá ser reproducido por la mayoría de los reproductores multimedia.

#### Método para grabar dentro de la videoconferencia

```
saveRecordedStream( stream, user ) {  
  let blob = new Blob( stream, { type: 'video/webm' } );  
  let file = new File( [blob], `${ user }-${ moment().unix() }-record.webm` );  
  saveAs( file );  
},
```

Figura 43. Estructura saveRecordedStream  
Elaborado por: Franklin Quiroga y Santiago Rosero

#### 3.4.2.4 *Compartir archivos*

Para completar el proceso de compartir archivos se ha utilizado dos “input”, el primero de tipo “file” que sirve para escoger el archivo que se desea compartir, mientras que el segundo de tipo “submit” se lo ha usado para poder subir dicho archivo al servidor. Como se visualiza en la Figura 45 todos los documentos compartidos se guardan en la siguiente dirección: “assets/upload/”, de acuerdo a la Figura 44 se utilizó el método fs.readdir para poder visualizar el contenido del directorio, entre sus argumentos se indica la ruta que contiene la información que se desea desplegar.

#### Método para mostrar todos los archivos que han subido los usuarios al servidor

```
app.get('/listadofotos',function listadofotos(req,res){  
  fs.readdir('assets/upload/',function (error,archivos){  
    var fotos='';  
    for(var x=0;x<archivos.length;x++) {  
      //fotos += '<br>';  
      fotos += '<a href="assets/upload/'+archivos[x]+'">'+archivos[x]+'</a><br>'  
    };  
    console.log(fotos);  
  }  
  res.writeHead(200, {'Content-Type': 'text/html'});  
  res.write('<!doctype html><html><head></head><body>'+fotos+'<a href="/"r</a>  
as</br></body></html>');  
  res.end();  
});  
});
```

Figura 44. Mostrar archivos subidos al servidor  
Elaborado por: Franklin Quiroga y Santiago Rosero

#### Método para subir un archivo al servidor

```
app.post('/index', function(req, res){
  if(req.files){
    var file = req.files.filename,
        filename = file.name;
    file.mv("assets/upload/"+filename, function(err){
      if(err){
        console.log(err);
        res.send("error ocured")
      }
      else{
        res.send("Done")
      }
    })
  }
})
})
```

Figura 45. Subir archivos

Elaborado por: Franklin Quiroga y Santiago Rosero

#### 3.4.2.5 Servidor Turn

El componente ICE contiene la información de configuración de la conectividad de WebRTC con un servidor, el método “getIceServer” solicita la información del servidor para el correcto funcionamiento de la videoconferencia.

#### Credenciales para la conexión con el servidor turn

```
getIceServer() {
  return {
    iceServers: [
      {
        urls: ["url_del_servidor_turn"]
      },
      {
        username: "Username_del_servidor_turn",
        credential: "Credencial_del_servidor_turn",
        urls: [
          "transport=udp",
          "transport=tcp"
        ]
      }
    ]
  }
}
```

Figura 46. Credenciales servidor turn

Elaborado por: Franklin Quiroga y Santiago Rosero

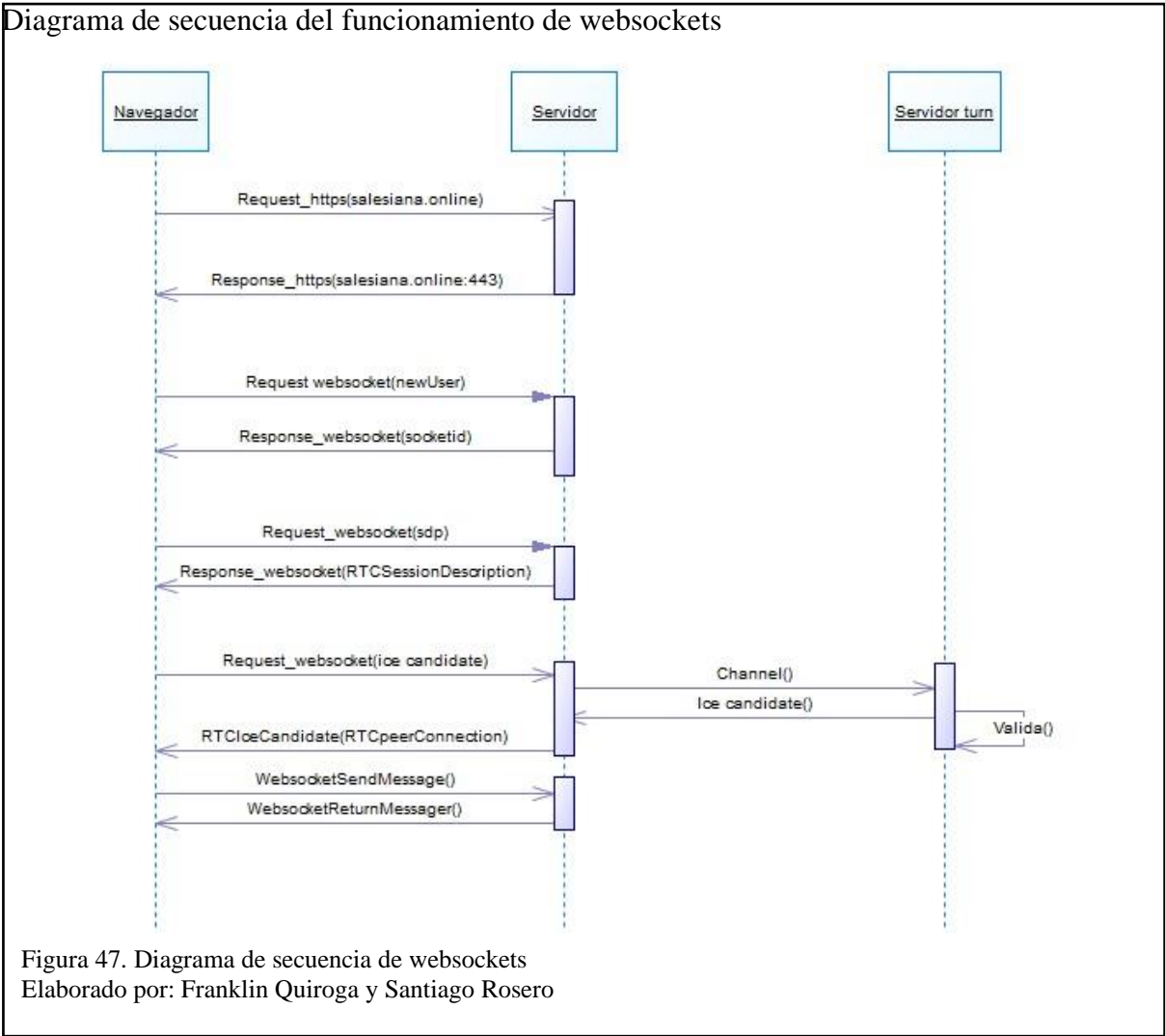


Para la mayoría de aplicaciones que utilizan el protocolo WebRTC necesitan de un servidor para retransmitir el tráfico, el motivo por el cual se integra un servidor turn es que los usuarios acceden a la plataforma desde diferentes ubicaciones.

Una vez que se tenga el servidor turn es necesario configurar RTCCOnfiguration, RTCPeerConnection y los parámetros de iceServers que son proporcionados por la empresa con la que se adquiere el servicio del servidor turn.

**3.4.2.6 Websocket**

Los websocket se ha utilizado para mantener la comunicación constante y en tiempo real entre el cliente y servidor. Para lo cual se ha tomado ayuda de la API de Websockets. El funcionamiento de websockets se lo detalla en la Figura 47.



Para su correcto funcionamiento se instaló “socket.io” como dependencia. Como se observa en la Figura 48 este tiene que recibir como parámetro la variable donde se inicializa el puerto y el resultado de lo antes mencionado se lo almacenará en la variable “io”, la misma que mantiene la conexión del socket con el sistema.

Dependencia socket.io
<pre>var io = require( 'socket.io' )( httpsServer );</pre>
Figura 48. Dependencia socket.io Elaborado por: Franklin Quiroga y Santiago Rosero

Como se muestra en la Figura 49 la primera acción que realiza el servidor es escuchar los eventos de aquellos usuarios que intentan conectarse al sitio web mediante el navegador.

Conexión del socket en la plataforma
<pre>socket.on( 'connect', () =&gt; {      socketId = socket.io.engine.id;</pre>
Figura 49. Conexión socket Elaborado por: Franklin Quiroga y Santiago Rosero

### 3.4.2.7 Chat

Para el desarrollo del chat se ha utilizado JavaScript tanto en el cliente como en el servidor, de igual manera con Websockets se mantendrá la comunicación en tiempo real y esto permitirá enviar mensajes a los usuarios que se encuentren conectados. En la Figura 50 se evidencia que mediante “socket.emit” se envía los mensajes y para que estos puedan ser observados por los demás participantes se llamará a la función “addChat”, el cual imprime el mensaje con el respectivo nombre del remitente y la fecha actual con la hora. Toda la información obtenida se la envía a través de la variable “infodiv”. El encargado de devolver los elementos a la interfaz gráfica es el método “appendChild”.

## Formato para visualizar los mensajes dentro de la plataforma

```
socket.on( 'chat', ( data ) => {
  h.addChat( data, 'remote' );
});
socket.on( 'chat hand', ( data ) => {
  h.addChatBut( data, 'remote' );
});
});

addChatBut( data, senderType ) {
  let chatMsgDiv = document.querySelector( '#chat-messages' );
  let contentAlign = 'justify-content-end';
  let senderName = 'You';
  let msgBg = 'bg-white';

  if ( senderType === 'remote' ) {
    contentAlign = 'justify-content-start';
    senderName = data.sender;
    msgBg = '';

    this.toggleChatNotificationBadge();
  }

  let infoDiv = document.createElement( 'div' );
  infoDiv.className = 'sender-info';
  infoDiv.innerHTML = `${ senderName } - ${ moment().format( 'Do MMMM, Y
YYY h:mm a' ) }`;

  let colDiv = document.createElement( 'div' );
  colDiv.className = `col-10 card chat-card msg ${ msgBg }`;
  colDiv.innerHTML = xssFilters.inHTMLData( data.msg ).autoLink( { target: "_blank", rel: "nofollow"});

  let rowDiv = document.createElement( 'div' );
  rowDiv.className = `row ${ contentAlign } mb-2`;

  colDiv.appendChild( infoDiv );
  rowDiv.appendChild( colDiv );
  chatMsgDiv.appendChild( rowDiv );

  if ( this.pageHasFocus ) {
    rowDiv.scrollIntoView();
  }
},
```

Figura 50. Estructura chat

Elaborado por: Franklin Quiroga y Santiago Rosero

### 3.5 Dimensionamiento

Las características mínimas que debe tener la máquina virtual donde se va a alojar la aplicación web son las siguientes:

Para el almacenamiento la máquina deberá tener 10 GB como mínimo, los cuales se repartirán de la siguiente manera, la carpeta donde se aloja el proyecto ocupa un total de 76MB, de los cuales 21MB corresponde al código, y el resto de espacio podría ser ocupado para implementar nuevas funcionalidades. Del total de almacenamiento restante no se deberá ocupar más del 75% de su capacidad máxima, esto con el objetivo de no interferir en el rendimiento de la máquina virtual.

El procesamiento máximo que se ha alcanzado utilizando la plataforma de videoconferencia en el servidor es del 20.39%, dejando un umbral de disponibilidad de 79.61%, esta información es obtenida a través de la herramienta de monitoreo de la Google Cloud. Por lo tanto, se podría decir que no es necesario utilizar más de un CPU en las características de una máquina virtual.

Para entablar la comunicación entre los usuarios dentro de la videoconferencia se ha utilizado el servidor turn server cloud “xirsys” el mismo que se comunicará con la plataforma de videoconferencia que se encuentra alojada en la máquina virtual de Google cloud. Esta comunicación se la realiza mediante la configuración del RTCPeerConnection el mismo que necesita la url, username y credential que son proporcionadas por la consola del servidor turn. La configuración realizada en el proyector se encuentra en el archivo “helpers.js” dentro de la carpeta “assets”.

Dos usuarios que utilizan la plataforma web realizan un consumo de 11.46 KiB/s, este valor corresponde cuando los usuarios empiezan a interactuar dentro del sistema. Para tener una idea del consumo que se va a llevar a cabo en el servidor turn se ha considerado los siguientes parámetros:

- ¿Cuántos usuarios espera en cada llamada?
- Aproximadamente, ¿cuánto durará cada llamada? (En minutos)
- ¿Aproximadamente cuántas llamadas por mes espera?

Después de haber analizado los parámetros mencionados, se ha determinado lo siguiente: dos usuarios que utilizan la plataforma tienen un consumo de 11.46 KiB/s, una videollamada en promedio durará 90 minutos equivalente a 5400 segundos y se espera tener 60 videollamadas al mes.

Dando como resultado:

X = consumo de 2 usuarios en KiB

$$x = 5.73 \text{ KiB/s} * 2$$

$$x = 11.46 \text{ KiB/s}$$

Y = consumo en KiB de 2 usuarios en una llamada de 90 minutos (5400 segundos)

$$y = 11.46 \frac{\text{KiB}}{\text{s}} * 5400 \text{ s}$$

$$y = 61884 \text{ KiB}$$

Z = consumo en KiB de 2 usuarios en 60 llamadas (al mes) con una duración de 90 minutos por llamada

$$z = 61884 \text{ KiB} * 60$$

$$z = 3713.040 \text{ KiB}$$

El resultado final es de 3713040 KiB lo que es igual a 3.80 GB de consumo de ancho de banda al mes en el servidor turn.

En base a este análisis el cálculo para el consumo de 40 estudiantes será de 152.09 GB al mes.

Al tener un alto consumo de ancho de banda se debe considerar en elegir un servidor que tenga las características apropiadas para realizar esta transferencia de tráfico.


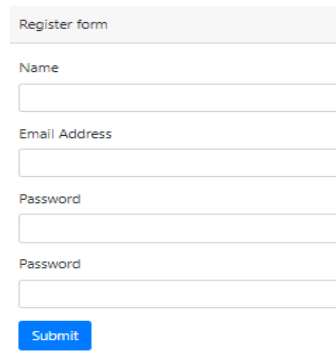
Como se puede observar la plataforma de videoconferencia utiliza un gran ancho de banda, por tal motivo es indispensable que al dimensionar las características del servidor turn se tome en consideración la capacidad que este permita para realizar este tipo de tráfico. Se debe también tener en cuenta que un servidor turn tiene grandes características, motivo por el cual los costos no son muy económicos sin embargo el beneficio será muy alto ya que permite garantizar las conexiones que el administrador del sistema permita.

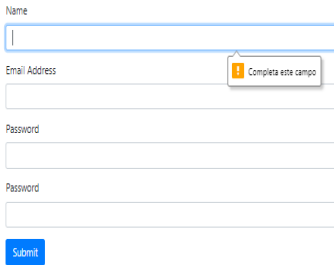
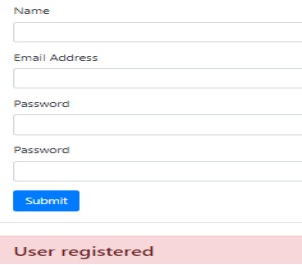
### 3.6 Pruebas

#### 3.6.1 Pruebas de caja negra

Permiten verificar el comportamiento y las funcionalidades del software acorde a las especificaciones funcionales que fueron descritas por el cliente en la etapa de análisis de requerimientos. En las siguientes tablas se va a especificar de manera detallada el estado de cada requerimiento funcional.


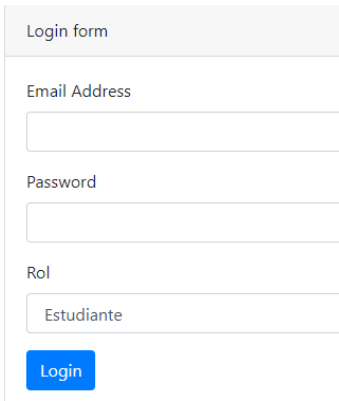
Tabla 29. Prueba de caja negra para Registro

Descripción	Estado	Observación
ACT1: Para acceder al sistema los usuarios deberán cumplir el formulario el registro.		
Los usuarios deberán pulsar el botón de “Register” para ingresar al módulo de registro	Aprobado	
El sistema muestra el formulario que debe de ser llenado para tener acceso al sistema	Aprobado	

Se debe tener en cuenta que cada campo tiene su validación correspondiente.	Aprobado	 <p>A screenshot of a registration form with fields for Name, Email Address, Password, and Password. A red tooltip with an exclamation mark icon and the text 'Completa este campo' is pointing to the Email Address field. A blue 'Submit' button is at the bottom.</p>
La plataforma notificará mediante un mensaje que el usuario ha sido registrado exitosamente.	Aprobado	 <p>A screenshot of a registration form with fields for Name, Email Address, Password, and Password. A blue 'Submit' button is at the bottom. Below the form, a pink banner displays the text 'User registered'.</p>


Nota: Prueba de caja negra para el registro de usuario en el sistema

Tabla 30. Prueba de caja negra para acceder a la plataforma

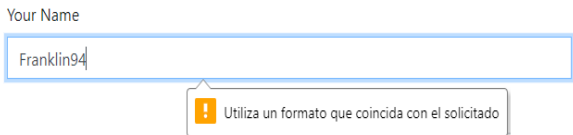
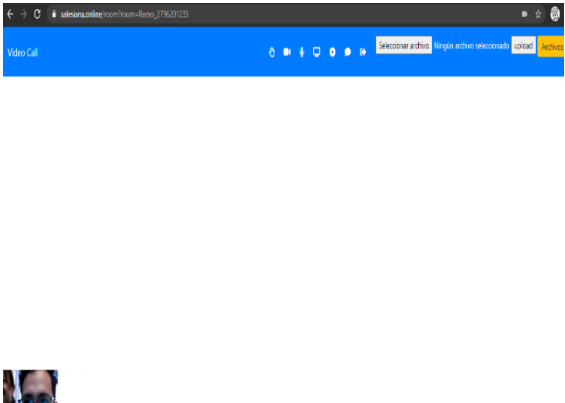
Descripción	Estado	Observación
ACT2: El sistema permitirá a los usuarios registrados ingresar a la plataforma		
El usuario deberá presionar el botón de “Login” para iniciar sesión.	Aprobado	 <p>A screenshot of a navigation menu with three items: 'Home', 'Login', and 'Register'. The 'Login' item is highlighted in green.</p>
El sistema mostrará un formulario que debe ser llenado para poder acceder a la plataforma.	Aprobado	 <p>A screenshot of a login form titled 'Login form'. It contains fields for Email Address, Password, and Rol. The Rol field has 'Estudiante' entered. A blue 'Login' button is at the bottom.</p>

Nota: Prueba de caja negra para que un usuario pueda ingresar a la plataforma

Tabla 31. Prueba de caja negra para acceder a una sala


Descripción	Estado	Observación
ACT3: Para ingresar al sistema de videoconferencia es necesario digitar el nombre y la respectiva contraseña proporcionada por el docente		
El usuario debe ingresar a “Profile” para acceder a la sala de videoconferencia.	Aprobado	
El usuario deberá ingresar en la caja de texto “Room” la información proporcionada por el docente y presionar el botón “Enter”.	Aprobado	
Para ingresar a la videoconferencia el usuario debe llenar el formulario con su nombre y contraseña de la sala.	Aprobado	

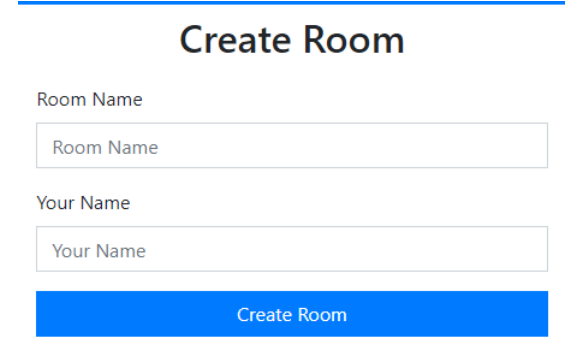



<p>El sistema validará que el nombre del usuario contenga solamente letras y una longitud máxima de 12 caracteres, este no debe tener números ni caracteres especiales.</p>	<p>Aprobado</p>	
<p>Si la información es ingresada correctamente el sistema permitirá ingresar al usuario a la sala de videoconferencia.</p>	<p>Aprobado</p>	

Nota: Prueba de caja negra para que un estudiante pueda acceder a una sala

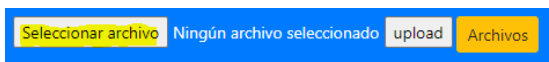
Tabla 32. Prueba de caja negra para crear una sala

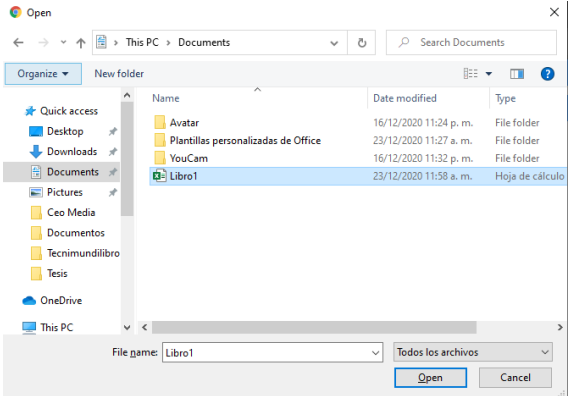
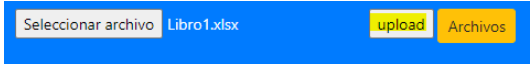
Descripción	Estado	Observación
<p>ACT4: El software permitirá crear salas, el docente tendrá la facultad de crear “n” cantidad de salas</p>		
<p>El docente debe ingresar a “Rooms” para crear una sala</p>	<p>Aprobado</p>	

Se deberá llenar los campos de Room Name y Your Name.	Aprobado	
Una vez llenado los campos se presionará el botón “Create Room”, el cual desplegará un mensaje.	Aprobado	
Una vez creada la sala el software habilitará un hipervínculo que redireccionará a la sala de videoconferencia.	Aprobado	<p>Room successfully created. Click <a href="#">here</a> to enter room. Share the room link with your partners.</p>

Nota: Prueba de caja negra para que un docente pueda crear una sala de videoconferencia

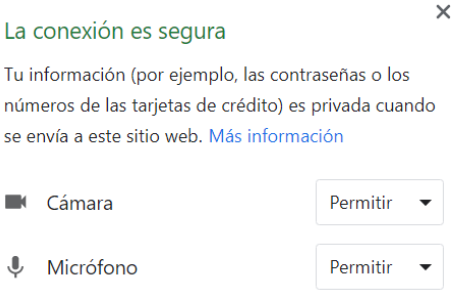
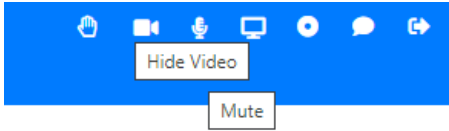
Tabla 33. Prueba de caja negra para compartir archivos

Descripción	Estado	Observación
ACT5: Se podrá compartir archivos en formatos específicos		
Para que el usuario pueda compartir información deberá dar clic en el botón	Aprobado	

“Seleccionar archivo”		
El sistema mostrará el Explorador de archivos el cual permitirá seleccionar el documento que se desea compartir.	Aprobado	
Para que el archivo sea subido al servidor se deberá presionar el botón “Upload”.	Aprobado	


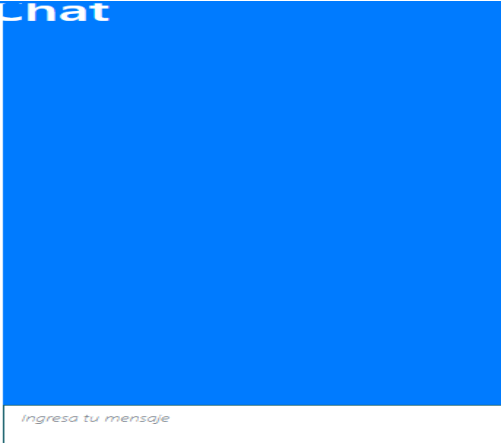

Nota: Prueba de caja negra del proceso de subir y compartir un archivo

Tabla 34. Prueba de caja negra para la interacción

Descripción	Estado	Observación
ACT6: El sistema permitirá silenciar el micrófono o apagar la cámara web		
Para que el usuario pueda interactuar con los demás participantes, este tiene la facultad de habilitar el micrófono y la cámara web	Aprobado	
El sistema contará con los botones “Hide video” y “Mute” para usar el audio y video.	Aprobado	

Nota: Prueba de caja negra para apagado/encendido de micrófono y/o cámara web

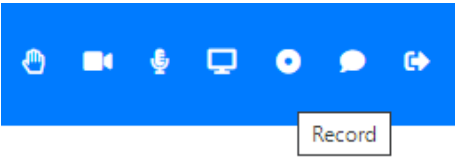
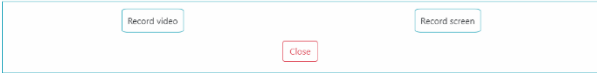

Tabla 35. Prueba de caja negra para el chat

Descripción	Estado	Observación
ACT7: El sistema permitirá una comunicación por medio de mensajes		
Para que el usuario pueda enviar mensajes deberá pulsar el botón “message”	Aprobado	
El sistema desplegará la caja de chat	Aprobado	
El usuario debe escribir el mensaje que desea enviar a los demás participantes y aplastar el botón “Enter” para que pueda ser visualizado.	Aprobado	

Nota: Prueba de caja negra para el envío de mensajes dentro de la videoconferencia


Tabla 36. Prueba de caja negra para grabar

Descripción	Estado	Observación
ACT8: Permitirá grabar cada clase que se realice por videoconferencia		

Para que el usuario pueda grabar la clase deberá usar el botón “Record”	Aprobado	
El usuario podrá elegir el tipo de grabación	Aprobado	
El usuario podrá finalizar la grabación cuando él lo desee	Aprobado	

Nota: Prueba de caja negra para grabar toda la videoconferencia actual

Tabla 37. Prueba de caja negra de Logout

Descripción	Estado	Observación
ACT9: El sistema deberá tener un botón que regrese al usuario al Home		
Si el usuario desea abandonar la videoconferencia, solo tendrá que oprimir el botón “Leave Room”	Aprobado	

Nota: Prueba de caja negra en el botón de “Logout”

### 3.6.2 Pruebas de aceptación (UAT)

Las pruebas de aceptación User Acceptance Testing se enfocan en comprobar que las especificaciones de los requerimientos funcionales se cumplan para lo cual estas pruebas fueron ejecutadas por los desarrolladores y el usuario para señalar diferentes observaciones y verificar si se cumplió el correcto funcionamiento.

Tabla 38. Pruebas de aceptación

<b>Código</b>	<b>Funcionalidad</b>	<b>Proceso</b>	<b>Observaciones</b>	<b>Estado</b>
RF1	Para acceder al sistema los usuarios deberán cumplir el formulario el registro	Ingresar a la siguiente dirección: <a href="https://salesiana.online">https://salesiana.online</a> Dirigirse a la ventana de “Register”	Debe tener un email valido Las contraseñas deben coincidir	OK
RF2	El sistema permitirá a los usuarios registrados ingresar a la plataforma	Ir a la ventana de “Login” Completar el formulario Clic en “Login”	Todos los campos tienen que ser completados	OK
RF3	Para ingresar al sistema de videoconferencia es necesario digitar el nombre y la respectiva contraseña proporcionada por el docente	Dirigirse a la ventana de “Profile” Ingresar el ID de la reunión Clic en “Ingreso” Llenar el formulario con el nombre y contraseña correspondiente Clic en “Enter Room”	La información de la sala es proporcionada por el docente	OK
RF4	El software permitirá crear salas, el docente tendrá la facultad de crear “n” cantidad de salas	Ir a la ventana “Room” Completar los campos Clic en “Create Room”	Solamente usuarios con el rol de Docente podrán crear salas	OK

RF5	Se podrá compartir archivos en formatos específicos	Clic en “examinar” Escoger el archivo a subir Presionar el botón “Upload”	Si se intenta subir un archivo no compatible con el sistema, este no se podrá visualizar en el servidor	OK
RF6	El sistema permitirá silenciar el micrófono o apagar la cámara web	Clic en “Hide video” para apagar la cámara web Clic en “Mute” para silenciar el audio	Siempre el sistema tendrá activado el micrófono y prendida la cámara	OK
RF7	El sistema permitirá una comunicación por medio de mensajes	Presiona el botón “Chat” Escribir el mensaje Presionar “Enter” para enviar el mensaje	Los mensajes podrán ser visualizados por todos los participantes de la sala	
RF8	Permitirá grabar cada clase que se realice por videoconferencia	Clic en el botón “Record” para empezar a grabar	Cuando un usuario finalice la grabación, esta se descargará inmediatamente	OK
RF9	El sistema deberá tener un botón que regrese al usuario al Home	Presionar el botón “Leave” para salir de la reunión	Al salir de la reunión se saldrá también del sistema	OK

Nota: Pruebas de aceptación en cada requerimiento funcional

### 3.6.3 Pruebas de carga

Para la prueba se consideró el siguiente escenario, en el cual se realizó 40 peticiones con un periodo de despliegue de 1 segundo. Esta prueba permitirá simular las conexiones de varios usuarios ingresando simultáneamente al Login y Register de la plataforma.

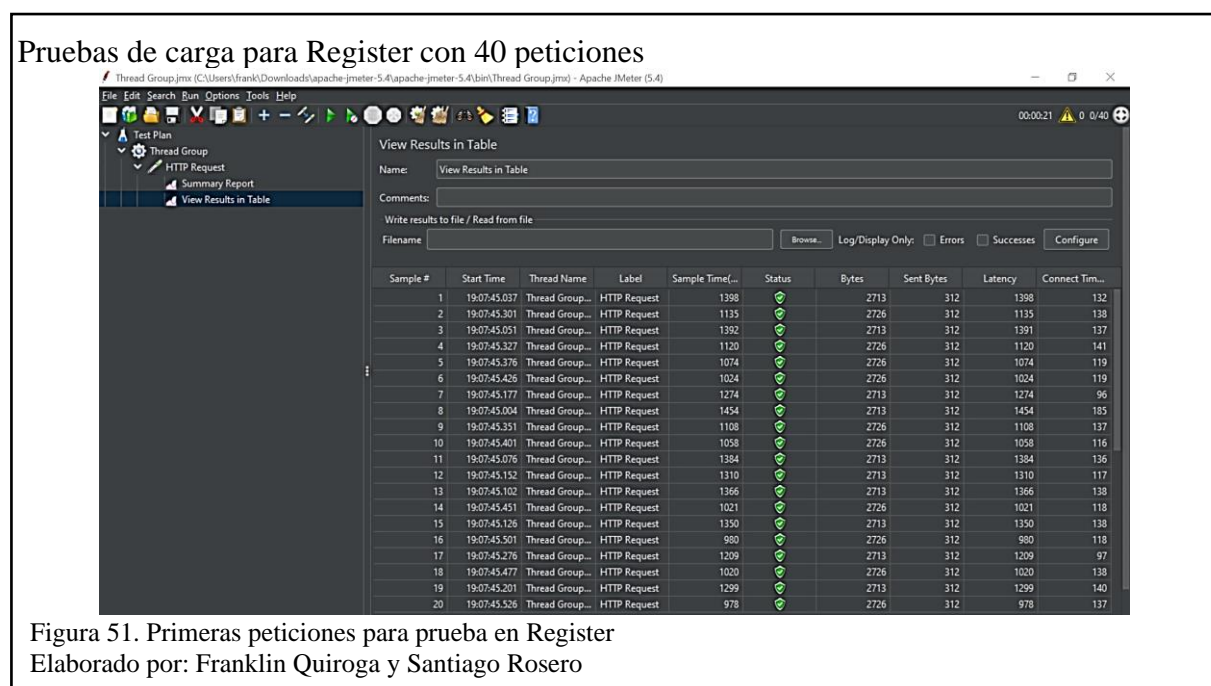
#### Register

Tabla 39. Pruebas de carga de Register

Label	Samples	Average	Min	Max	Received (KB/seg)	Sent (KB/seg)	Error %
HTTP POST	<b>40</b>	<b>1011</b>	<b>594</b>	<b>1454</b>	<b>67.86</b>	<b>7.78</b>	<b>0%</b>

Nota: Resultados obtenidos en las pruebas de carga para el módulo Register

Se ha realizado la prueba de carga al método HTTP post del módulo Register en este escenario se realizaron 40 peticiones como se evidencia en la Figura 51, de modo que se muestra en la Tabla 39 el tiempo mínimo en segundos es de 0.594 y el tiempo máximo en segundos es de 1.454 logrando un promedio de respuesta de 1.011 segundos. Los datos enviados son 7.78 KB por segundo y recibidos son 67.86 KB por segundo. Logrando un porcentaje de 0 errores lo cual indica que todas las peticiones fueron atendidas de manera exitosa. Los parámetros enviados para dichas pruebas se los muestra en la Figura 52.







## Login

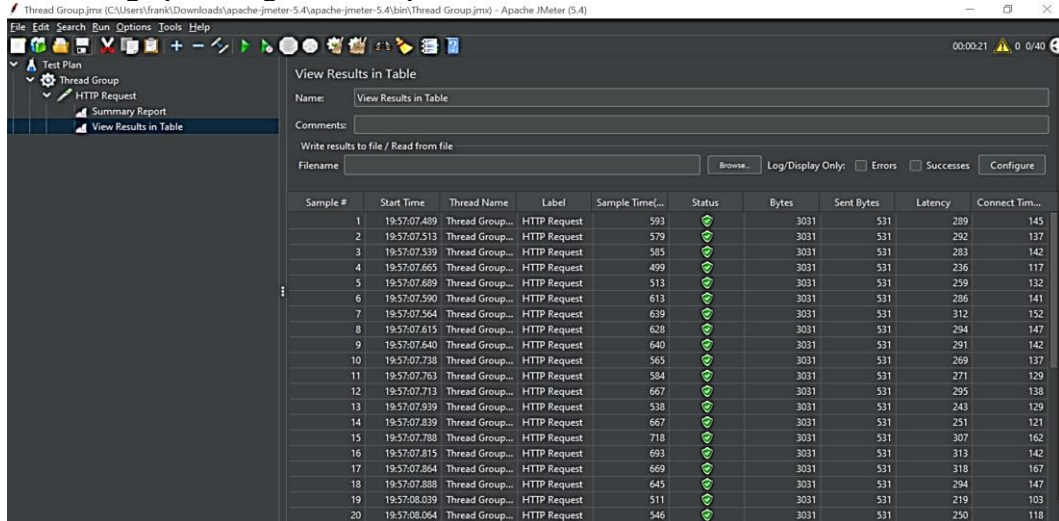
Tabla 40. Pruebas de carga de Login

Label	Samples	Average	Min	Max	Received (KB/seg)	Sent (KB/seg)	Error %
HTTP POST	40	647	499	786	68.64	12.02	0%

Nota: Resultaos obtenidos en las pruebas de carga para el módulo Login

Se ha realizado la prueba de carga al método HTTP post del módulo Login, como se evidencia en la Figura 53 se realizaron 40 peticiones, así mismo, de modo que se evidencia en la Tabla 40 el tiempo mínimo en segundos es de 0.499 y el tiempo máximo en segundos es de 0.786 logrando un promedio de respuesta de 0.647 segundos. Los datos enviados son 12.02 KB por segundo y recibidos son 68.64 KB por segundo. Logrando un porcentaje de 0 errores lo cual indica que todas las peticiones fueron atendidas de manera exitosa. Los parámetros para la respectiva prueba se muestran en la Figura 54.

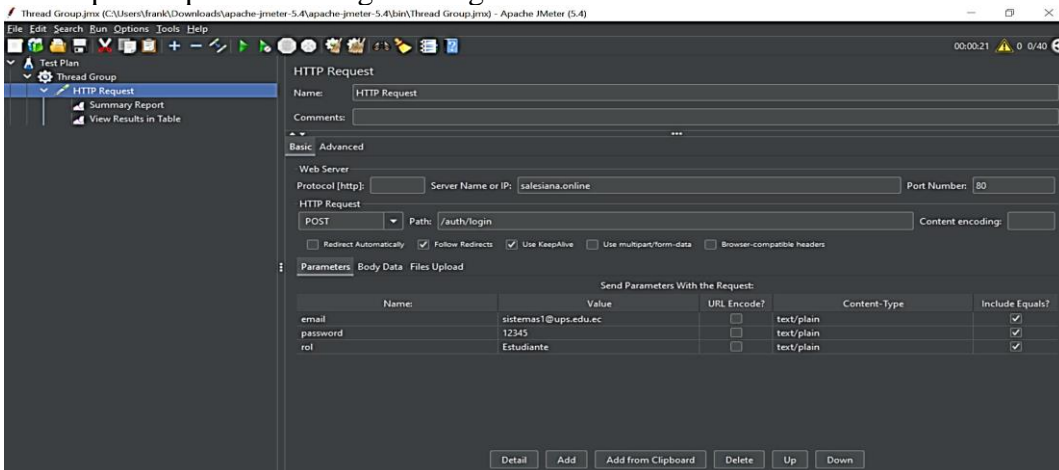
## Pruebas de carga para Login con 40 peticiones



Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latency	Connect Tim...
1	19-57:07.489	Thread Group...	HTTP Request	593	✓	3031	531	289	145
2	19-57:07.513	Thread Group...	HTTP Request	579	✓	3031	531	292	137
3	19-57:07.539	Thread Group...	HTTP Request	585	✓	3031	531	283	142
4	19-57:07.665	Thread Group...	HTTP Request	499	✓	3031	531	236	117
5	19-57:07.689	Thread Group...	HTTP Request	513	✓	3031	531	259	132
6	19-57:07.590	Thread Group...	HTTP Request	613	✓	3031	531	286	141
7	19-57:07.564	Thread Group...	HTTP Request	639	✓	3031	531	312	152
8	19-57:07.615	Thread Group...	HTTP Request	628	✓	3031	531	294	147
9	19-57:07.640	Thread Group...	HTTP Request	640	✓	3031	531	291	142
10	19-57:07.738	Thread Group...	HTTP Request	565	✓	3031	531	269	137
11	19-57:07.763	Thread Group...	HTTP Request	584	✓	3031	531	271	129
12	19-57:07.713	Thread Group...	HTTP Request	667	✓	3031	531	295	138
13	19-57:07.939	Thread Group...	HTTP Request	538	✓	3031	531	243	129
14	19-57:07.839	Thread Group...	HTTP Request	667	✓	3031	531	251	121
15	19-57:07.788	Thread Group...	HTTP Request	718	✓	3031	531	307	162
16	19-57:07.815	Thread Group...	HTTP Request	693	✓	3031	531	313	142
17	19-57:07.864	Thread Group...	HTTP Request	669	✓	3031	531	318	167
18	19-57:07.888	Thread Group...	HTTP Request	645	✓	3031	531	294	147
19	19-57:08.039	Thread Group...	HTTP Request	511	✓	3031	531	219	103
20	19-57:08.064	Thread Group...	HTTP Request	546	✓	3031	531	250	118

Figura 53. Primeras peticiones para prueba en Login  
Elaborado por: Franklin Quiroga y Santiago Rosero

## Parámetros para la prueba de carga en Login



HTTP Request

Name: HTTP Request

Comments: ---

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: salesiana.online Port Number: 80

HTTP Request

POST Path: /auth/login Content encoding: ---

Redirect Automatically  Follow Redirects  Use KeepAlive  Use multipart/form-data  Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name	Value	URL Encode?	Content-Type	Include Equals?
email	sistemas@ups.edu.ec	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>
password	12345	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>
rol	Estudiante	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>

Detail Add Add from Clipboard Delete Up Down

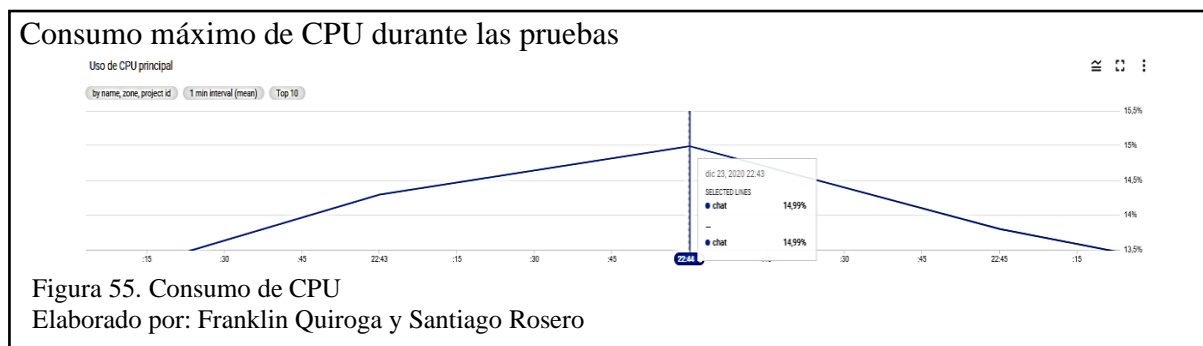
Figura 54. Parámetros para la ejecución de la prueba en Login  
Elaborado por: Franklin Quiroga y Santiago Rosero

### 3.6.4 Pruebas de rendimiento

Al ejecutar la siguiente prueba se ha creado 3 salas, cada una con 10 usuarios, las actividades que cada usuario realizaron fueron: inicio de sesión, acceder a la reunión de videoconferencia en la cual cada usuario ejecutó diferentes actividades como enviar mensajes, grabar la pantalla, compartir archivos, activar/desactivar audio/video, compartir pantalla y cierre de sesión. Para saber cuál es el consumo de la CPU, disco y red se utilizará el dashboard Monitoring de Google Cloud el cual presenta los datos en tiempo real de la plataforma.

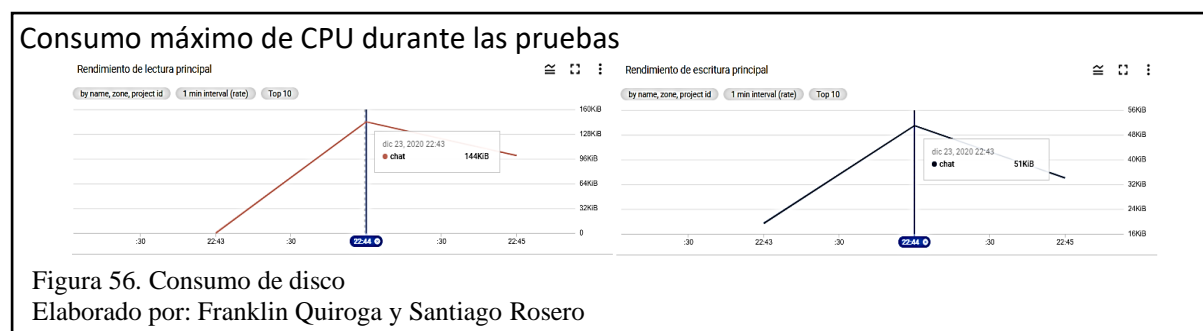
## CPU

Durante las pruebas ejecutadas se observó que el consumo máximo de CPU fue del 14.99%, como se observa en la Figura 55 este valor corresponde a la creación de 3 salas diferentes con una interacción de 10 participantes en cada una de ellas.



## Uso de disco

En las mismas pruebas ejecutadas se apreció que el rendimiento máximo de lectura en el disco fue de 144 KiB, mientras que el mayor rendimiento de escritura fue de 51 KiB, como se observa en la Figura 56 dichos valores corresponde a la interacción de 10 participantes en 3 salas de videoconferencia.



## Consumo de red

Para calcular el consumo de red se ha tomado en cuenta la tasa de transmisión que genera la plataforma, obteniendo como resultado el máximo tráfico enviado de 6.2 KiB y el mayor tráfico recibido de 5.64 KiB, como se aprecia en la Figura 57 cada valor corresponde a la interacción de 10 estudiantes en 3 diferentes salas de videoconferencia.



Tabla 41. Consumo de CPU por operación

Operación	Consumo de CPU	Duración
Inicio de sesión	12.54%	22:30 – 22:36
Reunión de videoconferencia	14.99%	22:36 – 23:01
Cierre de sesión	12.87%	23:01 – 23:06

Nota: Consumo de CPU en porcentaje de acuerdo con cada operación del sistema

## GLOSARIO

- **KiB:**

Unidad de almacenamiento de información o memoria interna que es equivalente a  $2^{10}$  bytes.

- **WebRTC:**

Comunicación en tiempo real, es de código abierto y fue desarrollado para solucionar problemas de comunicación.

- **Dependencia:**

Es una clase que depende de otra para realizar su funcionamiento.

- **NPM:**

Es un repositorio utilizado para la publicación de proyectos Nodejs de código abierto.

- **Handlebar**

Es una plantilla que permite separar el código html del javascript, este da como resultado la creación de un código más limpio y sencillo.

- **Sockets:**

Es un canal de comunicación bidireccional que realiza un conjunto de peticiones.

- **Pruebas UAT:**

Son pruebas de aceptación de usuarios que garantizan que el producto cumpla con las expectativas del cliente.

- **GNU:**

Sistema operativo de Unix que cuentan con una colección de programas.

- **TCP/IP:**

Conjunto de estándares que permiten a los equipos comunicarse en una red como internet.

- **UML:**

Es un modelo que permite crear diagramas y esquemas para el desarrollo de un software.

- **SSL:**

Es un estándar para mantener segura una conexión a internet.

- **HANDLEBARS:**

Plantilla que permite crear código html de una manera más fácil.

- **BACK END:**

Parte del código que los usuarios no pueden tener acceso, entre ellos se encuentran servidores, la aplicación y bases de datos.

- **FRONT END:**

Parte de una aplicación que interactúa con el usuario, es conocida como lado del cliente.

- **SQL:**

Lenguaje diseñado para almacenar información y gestionarla en una base de datos.

## CONCLUSIONES

- Se debe tener en cuenta que para la construcción de la plataforma web se realizó un análisis en base a la encuesta realizada por parte de los docentes y estudiantes, esta información ayudo a definir los requerimientos y el comportamiento del sitio web.
- Dentro del diseño se cree importante tener en consideración que nuestro sitio web se va a alojar en una cloud debido a las características y beneficios como son la alta disponibilidad y escalabilidad, de igual manera es importante recalcar que se puede reducir y controlar los costos debido a que la mayoría de clouds tienen integrado un dashboard de facturación.
- Así mismo, se determinó usar en el front end html y handlebar para desarrollar nuestra página web, bootstrap para estilizar el contenido junto con css y javascript para tomar las solicitudes de los usuarios, donde el back end está conformado por nodejs el cual es ejecutado en el lado del servidor y el framework utilizado es express que permite tener una estructura ya definida.
- En relación a lo antes expuesto, la plataforma de videoconferencia utilizo el protocolo WebRTC, para entablar una comunicación entre los participantes se ha utilizado RTCpeerConnection el mismo que permite utilizar el audio y video de cada ordenador.
- De esta manera se añadieron funcionalidades que permitan al usuario manejar la plataforma de una manera más intuitiva, para esto se implementaron botones que realicen funciones específicas de esta manera mejorar la experiencia de usuario mediante una interfaz personalizada y adaptable logrando optimizar el número de clicks para culminar una tarea.
- Al momento de realizar las pruebas correspondientes al desarrollo de software se utilizó el Monitoring de Google cloud para medir el rendimiento de la aplicación, así mismo se usó el software JMeter que permitió conocer la carga operativa de nuestra plataforma

a través del método POST, arrojando resultados positivos dando a entender que las tecnologías utilizadas son las correctas para la ejecución de este proyecto.

- Frente a las evidencias recaudadas se pudo inferir que las características específicas que debe tener tanto la cloud como el servidor turn son las siguientes: la capacidad de almacenamiento de la MV es de 10Gb, para su rendimiento se necesita 1 CPU y el ancho de banda del servidor turn no debe ser menos de 150 Gb.
- El uso de la metodología XP en el transcurso del desarrollo del proyecto se logró obtener una comunicación constante entre el equipo de desarrollo logrando una planificación adecuada para cumplir con los objetivos propuestos en el tiempo definido, de esta manera poder alcanzar el objetivo final a pesar de que hayan existido solicitudes de cambio en el durante la realización del sistema
- Finalmente, en comparación a otras soluciones de videoconferencia nuestro sistema resulto tener un costo aceptable, si la universidad desearía poner en producción a la plataforma la inversión seria mínima y la rentabilidad alta ya que al contar con una solución propia esta se puede adaptar a las necesidades de los usuarios de manera más rápida y personalizada.



## RECOMENDACIONES

- Si la universidad optara por utilizar la plataforma de videoconferencia en lugar de otras soluciones se recomienda utilizar los manuales que se encuentran cargados en la plataforma para que puedan conocer sobre el uso de las herramientas que contiene el sistema web.
- Para futuras implementaciones en el desarrollo web se recomienda utilizar nodejs en el back end para que la estructura del proyecto se mantenga y la implementación de nuevos módulos no sea complicado de realizar.
- Se recomienda que en futuras versiones se integre una base de datos relacional donde los usuarios puedan registrarse a la plataforma por materia, ya que esto no fue contemplado en los requerimientos actuales.
- Para mejorar la confidencialidad e integridad del proyecto sería recomendable integrar una autenticación 2FA, ya que en el presente proyecto solo se cuenta con los mecanismos de seguridad como la encriptación de contraseña y protocolo SSL.
- Si el proyecto será albergado en una máquina virtual del data center de la Universidad Politécnica Salesiana es importante tener desactivado los puertos 443 y 80, los mismo que son necesarios para entablar la comunicación entre los distintos usuarios.
- Se recomienda realizar un mantenimiento del software cada cierto tiempo debido a que ciertas herramientas utilizadas en el desarrollo podrían dejar de funcionar o necesitarán ser actualizadas a versiones más recientes y que a su vez sean compatibles.

## LISTA DE REFERENCIAS

- ATOM. (2020). *flight-manual.atom.io*. Recuperado el 24 de Junio de 2020, de Personalización Básica: <https://flight-manual.atom.io/using-atom/sections/basic-customization/>
- Fernández, J. (2013). *Introducción a las metodologías ágiles Otras formas de analizar y desarrollar*. Obtenido de exabyteinformatica: [https://www.exabyteinformatica.com/uoc/Informatica/Tecnicas\\_avanzadas\\_de\\_ingenieria\\_de\\_software/Tecnicas\\_avanzadas\\_de\\_ingenieria\\_de\\_software\\_\(Modulo\\_3\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnicas_avanzadas_de_ingenieria_de_software/Tecnicas_avanzadas_de_ingenieria_de_software_(Modulo_3).pdf)
- FileZilla. (12 de Junio de 2020). *Características de FileZilla*. Obtenido de filezilla-project.org: <https://wiki.filezilla-project.org/Documentation>
- Google. (2020). *Compute Engine*. Recuperado el 26 de Junio de 2020, de cloud.google.com: [https://cloud.google.com/compute?utm\\_campaign=latam-LATAM-all-es-dr-bkws-all-all-trial-b-dr-1008075-LUAC0009276&utm\\_term=KW\\_%2Bgoogle+%2Bvm-ST\\_%2BGoogle+%2BVM&gclid=CjwKCAjwltH3BRB6EiwAhj0IUOIFY71\\_g2dF-t2-uukJZTCCxVt\\_L-VoLpwt1hnWv9zbMKrKdpn6aBoCi9QQAxD\\_Bw](https://cloud.google.com/compute?utm_campaign=latam-LATAM-all-es-dr-bkws-all-all-trial-b-dr-1008075-LUAC0009276&utm_term=KW_%2Bgoogle+%2Bvm-ST_%2BGoogle+%2BVM&gclid=CjwKCAjwltH3BRB6EiwAhj0IUOIFY71_g2dF-t2-uukJZTCCxVt_L-VoLpwt1hnWv9zbMKrKdpn6aBoCi9QQAxD_Bw)
- Ubuntu. (2020). *Guía del servidor de Ubuntu*. Recuperado el 26 de Junio de 2020, de help.ubuntu.com: <https://help.ubuntu.com/18.04/serverguide/index.html>
- Coronel, C., Morris, S., & Rob, P. (2011). *Base de datos: diseño, implementación y administración*. Cengage Learning Editores.
- Díaz, M. C. A. G. (2015). Bases de Datos. *Centro Cultural Itaca SC*.
- Eguéiluz Pérez, J. (2012). *Introducción a JAVASCRIPT*.
- Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript*. Marcombo.

- Grau, X. F., & Segura, M. I. S. (2008). Desarrollo orientado a objetos con UML. *Recuperado El, 1.*
- Hassan, Y., Martín Fernández, F. J., & Iazza, G. (2004). Diseño web centrado en el usuario: usabilidad y arquitectura de la información. *Hipertext. Net, 2.*
- Hayes, B. (2008). *Cloud computing*. ACM New York, NY, USA.
- Joskowicz, J. (2008). Reglas y prácticas en eXtreme Programming. *Universidad de Vigo, 22.*
- Js, N., & JS, N. (2016). Node. js. *Tradução de: SILVA, AG Disponivel Em.*
- Marini, E. (2012). El modelo cliente/servidor. *Recuperado El, 5.*
- Maza, M. Á. S. (2012). *Javascript*. Innovación y Cualificación.
- Medina, A. C. (2003). La videoconferencia: conceptualización, elementos y uso educativo. *Etic@ Net: Revista Científica Electrónica de Educación y Comunicación En La Sociedad Del Conocimiento, 2, 8.*
- Morales Salcedo, R., & others. (1999). Aplicaciones de la Videoconferencia en Bibliotecas Digitales. *REPOSITORIO NACIONAL CONACYT.*
- Pérez Pérez, M. J., & others. (2012). *Guía comparativa de Metodologías ágiles.*
- Rios, E. H. (2012). Arrancar con HTML5. *Arrancar Con HTML5, 5–6.*
- Sánchez Arroyo, M. E. (2001). Integración de la videoconferencia en la educación a distancia. *Pixel-Bit. Revista de Medios y Educación, 17, 89-98.*
- Santillán, L. A. C., Ginestà, M. G., & Mora, Ó. P. (2014). Bases de datos en MySQL. *Universitat Oberta de Catalunya.*
- Shieh, C.-Y. M. (2013). *Distributed firewall architecture using virtual machines*. Google Patents.

- Stark, J., Nathan, P., Papaconstantinou, J., Lagerstrom, P., & Hope, P. (2010). *Building Android apps with HTML, CSS, and JavaScript*. “O’Reilly Media, Inc.”
- Tobón, L. M. E., & Carmona, L. E. D. (2007). *Caso práctico de la metodología ágil XP al desarrollo de software*. Universidad Tecnológica de Pereira. Facultad de Ingenierías Eléctricas....
- Woodraska, D., Sanford, M., & Xu, D. (2011). Security mutation testing of the FileZilla FTP server. *Proceedings of the 2011 ACM Symposium on Applied Computing*, 1425–1430.
- AMAZON. (2021). Obtenido de aws.amazon.com: <https://aws.amazon.com/es/ec2/instance-types/>
- Shahriar, A. (2014). *Digital Video, Concepts, Methods, and Metrics*. California: Apress open.
- Azaustre, C. (2016). *Aprendiendo Javascript*. carlosazaustre.es.