

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas
Carrera de Ingeniería en Sistemas Computacionales

**DESARROLLO DE UN MÓDULO WEB DE VIDEOCONFERENCIAS COMO
COMPONENTE DEL ENTORNO VIRTUAL DE APRENDIZAJE INTEGRADO
(EVAI) PARA LA EMPRESA IEREC.**

Trabajo de grado previo a la obtención del título de ingeniero
en Sistemas Computacionales

Autor:

Pedro Jhoel Malte Andrade

Director:

Msc. Cosme Ortega

Ibarra – Ecuador

2022



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1003852553		
APELLIDOS Y NOMBRES:	MALTE ANDRADE PEDRO JHOEL		
DIRECCIÓN:	IBARRA, MIRADOR DEL ADUANA		
EMAIL:	pjmaltea@utn.edu.ec		
TELÉFONO FIJO:	N/A	TELÉFONO MÓVIL:	0989383795

DATOS DE LA OBRA	
TÍTULO:	DESARROLLO DE UN MÓDULO WEB DE VIDEOCONFERENCIAS COMO COMPONENTE DEL ENTORNO VIRTUAL DE APRENDIZAJE INTEGRADO (EVAI) PARA LA EMPRESA IREC.
AUTOR:	MALTE ANDRADE PEDRO JHOEL
FECHA: DD/MM/AAAA	17/06/2022
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TÍTULO POR EL QUE OPTA:	INGENIERIA EN SISTEMAS COMPUTACIONALES.
ASESOR /DIRECTOR:	MSC. COSME MACARTHUR ORTEGA BUSTAMANTE

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 20 días del mes de julio de 2022.

EL AUTOR:

.....
Nombre: Pedro Andrade

CERTIFICADO DEL DIRECTOR DE TRABAJO DE GRADO

UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN DEL DIRECTOR

Por medio del presente yo Ing. Cosme Ortega, MSc, certifico que el Sr. Pedro Jhoel Malte Andrade, portador de la cédula de ciudadanía Nro. 1003852553. Ha trabajado en el desarrollo del proyecto de tesis **“Desarrollo De Un Módulo Web De Videoconferencias Como Componente Del Entorno Virtual De Aprendizaje Integrado (EVAI) Para La Empresa IREC.”**, previo a la obtención del título de Ingeniería en Sistemas Computacionales, lo cual ha realizado en su total responsabilidad.

Es todo cuanto puedo certificar en honor a la verdad.

Atentamente:



Ing. Cosme Ortega, MSc
DIRECTOR DE TESIS

CERTIFICADO

Me permito informar a Ustedes que el señor **PEDRO JOEL MALTE ANDRADE**, con cédula de ciudadanía Nro. **100385255**. Estudiante de la Universidad Técnica del Norte, ha realizado su Trabajo de Grado con el tema: **“DESARROLLO DE UN MÓDULO WEB DE VIDEOCONFERENCIAS COMO COMPONENTE DEL ENTORNO VIRTUAL DE APRENDIZAJE INTEGRADO (EVAI) PARA LA EMPRESA IEREC.”** Cumpliendo con todos los requisitos reglamentarios de aprobación de la empresa, con cualidades de responsabilidad y profesionalismos.

Para efecto, se extiende el presente CERTIFICADO DE CULMINACIÓN DEL SOFTWARE, en la ciudad de Quito a los 3 días del mes de febrero de 2022.

Agradezco su atención.

Atentamente,



Ing. Guillermo Pérez Msc
CEO IER ECUADOR



0987 127 655
+593987127655



ier.energiasrenovables@gmail.com



Los Juncos 471 y Eloy Alfaro
Quito - Ecuador

Dedicatoria

El presente trabajo está dedicado a Dios, a mis padres Pedro Malte y Cristina Andrade que fueron la piedra angular en mi etapa de preparación académica, contentando con su apoyo moral, económico y emocional, por estar siempre pendiente de mí con su amor y sus consejos, brindándome su total apoyo y motivándome a seguir en el camino correcto y superar las dificultades presentes en el mismo. A mis amigos y compañeros de clases que también contribuyeron en mi formación y lograr conseguir mi meta.

Agradecimientos

Agradezco a Dios por darme la fuerza y la paciencia para alcanzar mis objetivos y por darme bellos momentos que compartí con mis seres queridos.

Gracias a mis padres y hermanos por darme todo su amor y motivarme para seguir adelante, agradezco por inculcarme buenos valores y darme la oportunidad de contar con una buena educación y llegar al camino del profesionalismo.

A mi tutor el MSc. Cosme Ortega a quién le agradezco por haberme apoyado con su conocimiento y gestión en el transcurso del desarrollo de mi trabajo de grado.

A mis familiares y docentes que fueron guía durante mi carrera, dándome su apoyo y compartiendo su conocimiento, ayudándome a crecer como persona y profesional.

Tabla de Contenido

Resumen	XIII
Introducción	XIV
Antecedentes.....	XIV
Situación Actual	XIV
Prospectiva	XV
Planteamiento del Problema	XV
Objetivos.....	XVI
1. Objetivo General	XVI
2. Objetivos Específicos	XVI
Alcance	XVI
Arquitectura	XVI
Metodología	XVII
CAPÍTULO 1.....	1
1. Marco Teórico.....	1
1.1. Software Libre.....	1
1.1.1. Tipos de software libre.....	1
1.2. Jitisi meet.....	1
1.2.1. Componentes	2
1.2.2. Diagrama de Arquitectura	3
1.3. Cloud Computing.....	4
1.3.1. Capas del Cloud Computing	4
1.3.2. Tipos de Cloud Computing.....	6
1.4. Bases de Datos no SQL	6
1.4.1. Bases de datos NoSQL	7
1.5. Metodología de Desarrollo de Software: Extreme Programming (XP).....	9
1.5.1. Objetivos de Extreme Programming	10
1.5.2. Etapas y ciclo de Extreme Programming	11
1.5.3. Roles de la Metodología Extreme Programming	12
1.6. Backend y Frontend.....	13
1.6.1. Node JS como servidor (Backend).....	13
1.6.2. React JS para creación de interfaces (Frontend)	14
CAPÍTULO 2.....	16
2. Desarrollo	16
2.1. Planificación.....	16
2.1.1. Metodología de Desarrollo.....	17

2.1.2.	Levantamiento de requerimientos	17
2.1.3.	Casos de Uso	18
2.1.4.	Historias de Usuario.....	20
2.1.5.	Iteraciones	20
2.2.	Diseño	23
2.2.1.	Arquitectura del sistema	23
2.2.2.	Diagrama de procesos.....	24
2.2.3.	Arquitectura base de datos	26
2.3.	Codificación	27
2.3.1.	Configuración del entorno de desarrollo.....	28
2.3.2.	Creación de variables de entorno	29
2.3.3.	Configuración y ejecución de la base de datos NoSQL de calendario	30
2.3.4.	Creación backend.....	32
2.3.5.	Creación de rutas en el backend.....	34
2.3.6.	Creación de servicios en el backend.....	35
2.3.7.	Configuración de Envío de correos en el backend	36
2.3.8.	Creación frontend	37
2.3.9.	Creación de componentes en el frontend.....	38
2.3.10.	Creación de reducers en el frontend	39
2.3.11.	Creación de rutas en el frontend	40
2.3.12.	Creación de interfaces en el backend y frontend.....	41
2.4.	Pruebas	42
2.4.1.	Pruebas de carga con Postman.....	42
2.4.2.	Pruebas unitarias	44
CAPÍTULO 3.....		47
3.	Resultados.....	47
3.1.	Validación de resultados mediante la ISO 25010.....	47
3.2.	Metodología	47
3.3.	Evaluación	48
3.4.	Análisis de resultados	49
3.5.	Interpretación de resultados.....	59
CONCLUSIONES		63
RECOMENDACIONES		64
REFERENCIAS		65
ANEXOS.....		67

Índice de Figuras

Fig. 1. Planteamiento del problema.....	XV
Fig. 2. Tecnologías.....	XVII
Fig. 3. Diagrama metodología	XVIII
Fig. 4. Arquitectura de Componentes de Jitsi.....	3
Fig. 5. Tipos de cloud	5
Fig. 6. Arquitectura node js.....	14
Fig. 7. Ciclo de vida de un componente	15
Fig. 8. Caso de uso administrador.....	18
Fig. 9. Caso de uso usuario.....	19
Fig. 10. Arquitectura del sistema	24
Fig. 11. Diagrama de proceso calendario	25
Fig. 12. Diagrama de proceso videoconferencia.....	26
Fig. 13. Estructura de datos JSON en mongoDB	27
Fig. 14. Modelo base de datos noSQL meeting	27
Fig. 15. Ejemplo de variables de entorno.....	29
Fig. 16. Variables de entorno jitsi	30
Fig. 17. Configuración base de datos mongoDB	31
Fig. 18. Ejecución y conexión base de datos mongoDB.....	31
Fig. 19. Ejecución comando creación backend.....	32
Fig. 20. Estructura de las carpetas backend.....	33
Fig. 21. Repositorio oficial jitsi.....	33
Fig. 22. Instalación paquetes jitsi.....	34
Fig. 23. Creación de peticiones http.....	35
Fig. 24. Método obtención de todas las videoconferencias	36
Fig. 25. Información correo.....	36
Fig. 26. Método envío de correo.....	37
Fig. 27. Ejecución comando creación frontend	37
Fig. 28. Estructura de las carpetas frontend.....	38
Fig. 29. Componente barra navegación	39
Fig. 30. Reducer de inicio y cierre de sesión	40
Fig. 31. Rutas login	41
Fig. 32. Interfaz meeting	42
Fig. 33. Colección http.....	43
Fig. 34. Inserción de URL para a consumir.....	43
Fig. 35. Resultado de petición http	44
Fig. 36. Configuración de jasmine pruebas unitarias	45
Fig. 37. Código para inicio de pruebas unitarias.....	45
Fig. 38. Resultado pruebas unitarias	46
Fig. 39. Resultados de la pregunta 1 de la encuesta SUS.....	50
Fig. 40. Resultados de la pregunta 2 de la encuesta SUS.....	51
Fig. 41. Resultados de la pregunta 3 de la encuesta SUS.....	52
Fig. 42. Resultados de la pregunta 4 de la encuesta SUS.....	53
Fig. 43. Resultados de la pregunta 5 de la encuesta SUS.....	54
Fig. 44. Resultados de la pregunta 6 de la encuesta SUS.....	55
Fig. 45. Resultados de la pregunta 7 de la encuesta SUS.....	56
Fig. 46. Resultados de la pregunta 8 de la encuesta SUS.....	57

Fig. 47. Resultados de la pregunta 9 de la encuesta SUS.....	58
Fig. 48. Resultados de la pregunta 10 de la encuesta SUS.....	59
Fig. 49. Grado de calificación SUS.....	62

Índice de Cuadros

TABLA 1. Partes del módulo de videoconferencias	16
TABLA 2. Roles Extreme Programming	17
TABLA 3. Descripción actor administrador	18
TABLA 4. Descripción actor usuario	19
TABLA 5. Listado historias de usuario.....	20
TABLA 6. Historia de usuario H1	20
TABLA 7. Historia de usuario H2.....	21
TABLA 8. Historia de usuario H3.....	21
TABLA 9. Historia de usuario H4.....	21
TABLA 10. Historia de usuario H5.....	22
TABLA 11. Historia de usuario H6.....	22
TABLA 12. Historia de usuario H7.....	22
TABLA 13. Historia de usuario H8.....	23
TABLA 14. Historia de usuario H9.....	23
TABLA 15. Software necesario.....	28
TABLA 16. Puntuación para escala de Likert.....	48
TABLA 17. Resultados de la encuesta por pregunta	49
TABLA 18. Valoración de las respuestas de escala SUS	59
TABLA 19. Puntuación individual de cada encuesta.....	60
TABLA 20. Sumatoria Preguntas pares e impares.....	61
TABLA 21. Promedio de la escala SUS	61

Resumen

El proyecto planteado planea resolver los problemas internos de la empresa IEREC tales como los altos niveles de inconformidad a causa planificación desorganizada en la planificación de las videoconferencias causando problemas tales como las pérdidas de clientes, la empresa no cuenta con un software que ayude automatizar totalmente los procesos académicos en los que la empresa se desempeña.

Para automatizar y mejorar los procesos se optó por el desarrollo de un módulo web de videoconferencias como componente de un entorno virtual de aprendizaje integrado (EVAI). La creación del software se inició realizando un levantamiento de requerimientos donde se define las necesidades de la empresa y posteriormente se desarrolló e implementó el módulo web.

Para el desarrollo del módulo web se utilizó JavaScript como lenguaje base de programación, donde se utilizó node JS para la construcción del backend con servicios REST y react JS para la creación de interfaces para la para el almacenamiento de datos se utilizó mongoDB.

Para validar el software se evaluó la Usabilidad mediante la ISO 25010 a través de un cuestionario SUS, del cual se obtuvo como resultado una calificación de 86.67, que representa un grado B que significa que la usabilidad del software es buena y aceptable.

Introducción

Antecedentes

La empresa fue creada para dar capacitaciones y servicios técnicos sobre recursos renovables con el pasar del tiempo la empresa fue creciendo y los estudiantes aumentando, generando un problema de falta de organización y automatización de procesos

Las video conferencias desde que se fundó la empresa fueron dados en la aplicación de videoconferencias de Google meet y con un constante crecimiento de los alumnos, las conferencias al comenzar a llegar a más gente comenzaron a generar inconvenientes y se producía pérdida de información en los registros de las videoconferencias ya que los procesos manuales tomaban más tiempo y entorpecen el proceso.

La empresa IERec, dedicada a buscar soluciones y exportar conocimiento sobre el uso eficiente de las energías renovables, trata de encontrar nuevos modelos de negocio que les ayude a crecer como empresa, desde el momento de su fundación la empresa ha tenido un gran éxito que se llega a un punto en de mejorar automatizar procesos. Hasta

la presente fecha el desarrollo de toda la gestión de los procesos de su negocio se ha realizado de manera manual.

Situación Actual

La empresa se dedica a dar capacitaciones y asistencia técnica al no tener una plataforma que una todo los módulos en general esta no puede ofertar cursos formalmente donde pueda evidenciar los servicios que esta brinda y que avalan la seguridad de la empresa se ven forzados a usar plataformas externas como Facebook, Zoom o Meet el cual también es un problema porque las personas tanto estudiantes como profesores deben aprender varias tecnologías que hacen perdiendo interacción con la empresa y la esta no es tomada de una manera seria comprometiendo la integrada de sí misma.

El modelo de negocio para informar sobre las conferencias era el uso de WhatsApp y Facebook sirviéndose de estos servicios para detallar información sobre horarios y realizar planeación de conferencias, la información es desordenadas y por lo cual tanto los estudiantes como la empresa terminan gastando demasiados esfuerzos en realizar registros

de los horarios perdiendo tiempo y la empresa generando altos niveles de inconformidad en los estudiantes.

Prospectiva

Se quiere desarrollar el módulo videoconferencias ya que la empresa al tener un proceso manual de registro y de organización de conferencias esta pierde información y el procesamiento de esta es muy lenta. El módulo de videoconferencias pretende automatizar y mejorar el proceso por medio de tecnologías actuales, en el backend ayudaran a que toda la información de registros sea procesada rápidamente y mostrada en el frontend de manera los datos generen valor y aumente la credibilidad y el nivel de interacción con la empresa IERec. También se hará el uso de servidores que ya estan preparados para software de videoconferencias dando una mayor robustez al desarrollo y mejora del módulo.

Planteamiento del Problema

Elevados niveles de inconformidad por una planificación desorganizada en la planificación de las videoconferencias incidiendo en la pérdida de clientes.

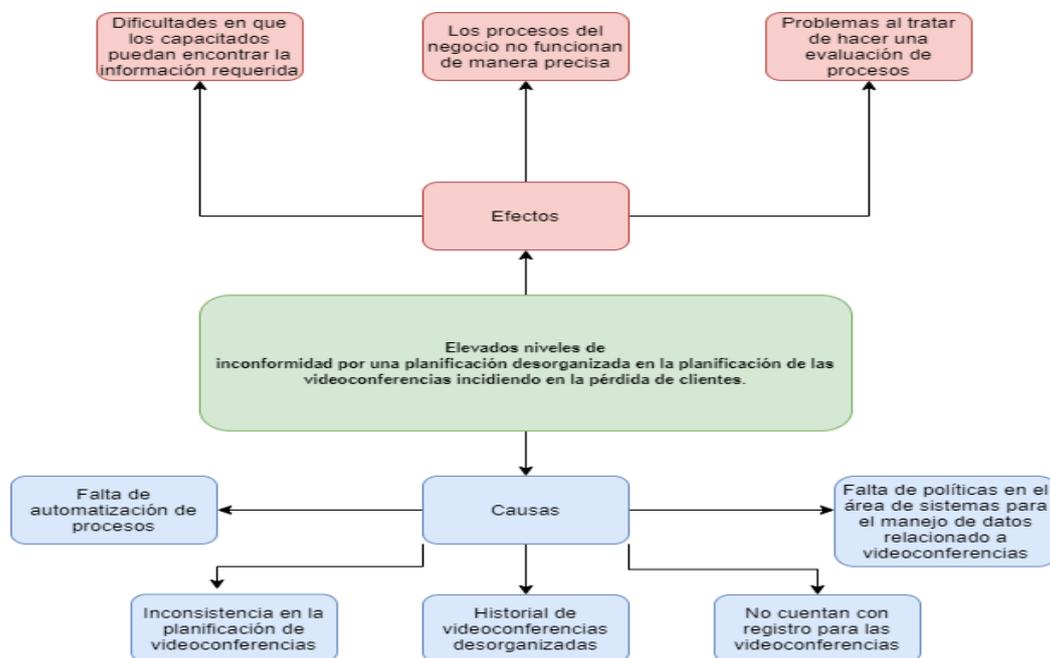


Fig. 1. Planteamiento del problema

Fuente: Propia

Objetivos

1. Objetivo General

- Desarrollar un módulo web para mejorar y fortalecer los procesos académicos de aprendizaje.

2. Objetivos Específicos

- Definir marco teórico sobre los componentes de la plataforma de videoconferencias jitsi y la funcionalidad del software.
- Implementación del módulo videoconferencias sobre plataformas web.
- Validación del software utilizando la ISO 25010 con la subcaracterística de usabilidad.

Alcance

Lograr implantar con éxito el módulo propuesto que este en conjunto a los objetivos del negocio este constara de componentes en un servidor para tener a disponibilidad las videoconferencias.

Se incorporará un gestor de actividades donde facilite al administrador la creación de conferencias siguiendo el modelo de negocio de la empresa, el administrador podrá crear, visualizar editar y eliminar las videoconferencias programadas.

Arquitectura

El software será construido con el modelo MVC y con microservicios que ayudaran a dar mayor velocidad de respuesta para mostrar más rápido información en pantalla.

Para la integración y construcción de este módulo se usará las siguientes tecnologías

- JavaScript como lenguaje base
- Un runtime de JavaScript NodeJS para la creación del backend
- Framework ReactJS para la construcción del Frontend
- Tecnología de estados ReduxJs
- Versionamiento con git y github
- Integración de TypeScript para JavaScript para un código tipado
- Express como framework de construcción del Backend

- Base de datos no relacional MongoDB.

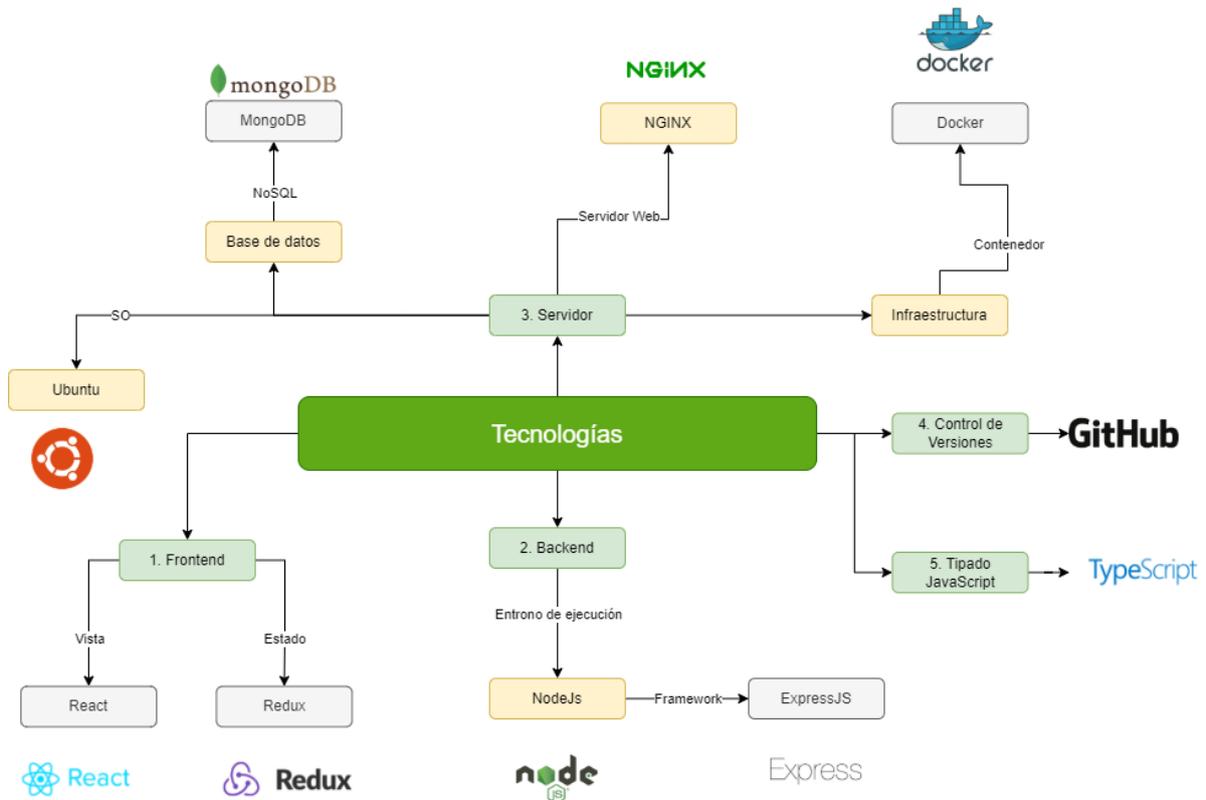


Fig. 2. Tecnologías

Fuente: Propia

Metodología

Se hará uso del método ingenieril como la estrategia para lograr el mejor cambio, con tecnologías actuales, planteando además que esta estrategia evidente que está asociada al uso de heurístico como una técnica o procedimiento práctico o informal, para resolver problemas.



Fig. 3. Diagrama metodología

Fuente: Propia

CAPÍTULO 1

1. Marco Teórico

1.1. Software Libre

El software libre explícitamente se refiere a la libertad que el usuario final tiene para ejecutar programas sea cual sea el índole o propósito, a continuación, se detalla las 4 libertades fundamentales con los que el usuario trabaja en el software libre. propósito de uso común.(Castro, 2014)

- Libertad para distribución de copias sin restricción alguna.
- Libertad para estudio y modificación. (Requisito: código fuente disponible).
- Libertad para publicar el programa mejorado. (Requisito: código fuente disponible)

1.1.1. Tipos de software libre

Stallman (2020) define al software libre como libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Nos presenta especialmente a cuatro clases de libertad para los usuarios de software:

- Libertad 0: la libertad para ejecutar el programa sea cual sea nuestro propósito.
- Libertad 1: la libertad para estudiar el funcionamiento del programa y adaptarlo a tus necesidades el acceso al código fuente es condición indispensable para esto.
- Libertad 2: la libertad para redistribuir copias y ayudar así a tu vecino.
- Libertad 3: la libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad el acceso al código fuente es condición indispensable para esto.

1.2. Jitisi meet

Jitisi es un proyecto financiado por la empresa Atlassian, con el que se desarrolla un set de herramientas gratuitas y de código abierto con las que poder realizar videoconferencias. Así pues, detrás de este nombre se esconden varias alternativas a las

aplicaciones de videollamadas más populares que llevan desde 2003 centrándose en la seguridad que da el que cualquiera pueda revisar su código. Jitsi es una colección de proyectos y librerías de código abierto que brindan capacidades de videoconferencia de vanguardia que son seguras, fáciles de usar.(Introduction · Jitsi Meet Handbook, n.d.)

1.2.1. Componentes

Según página Jitsi Meet Handbook (n.d.), Jitsi comprende una colección de proyectos y librerías:

- Jitsi Meet: Es una aplicación que proporcion videoconferencias escalables y de alta calidad. Aprovechando librerías de React y React Native.
- Jitsi Videobridge (JVB): servidor compatible con WebRTC diseñado para enrutar transmisiones de video entre los participantes en una conferencia.
- Jitsi Conference Focus (jicofo): componente de enfoque del lado del servidor utilizado en las conferencias Jitsi Meet que gestiona las sesiones de medios y actúa como equilibrador de carga entre cada uno de los participantes y el puente de vídeo.
- Jitsi Gateway to SIP (jigasi): aplicación del lado del servidor que permite a los clientes SIP habituales unirse a conferencias Jitsi Meet.
- Jitsi Broadcasting Infrastructure (jibri): conjunto de herramientas para grabar y transmitir una conferencia Jitsi Meet que funciona iniciando una instancia de Chrome renderizada en un framebuffer virtual y capturando y codificando la salida con ffmpeg.

1.2.2. Diagrama de Arquitectura

Las conexiones individuales entre los componentes, así como sus integraciones externas, se describen en la Fig. 4.

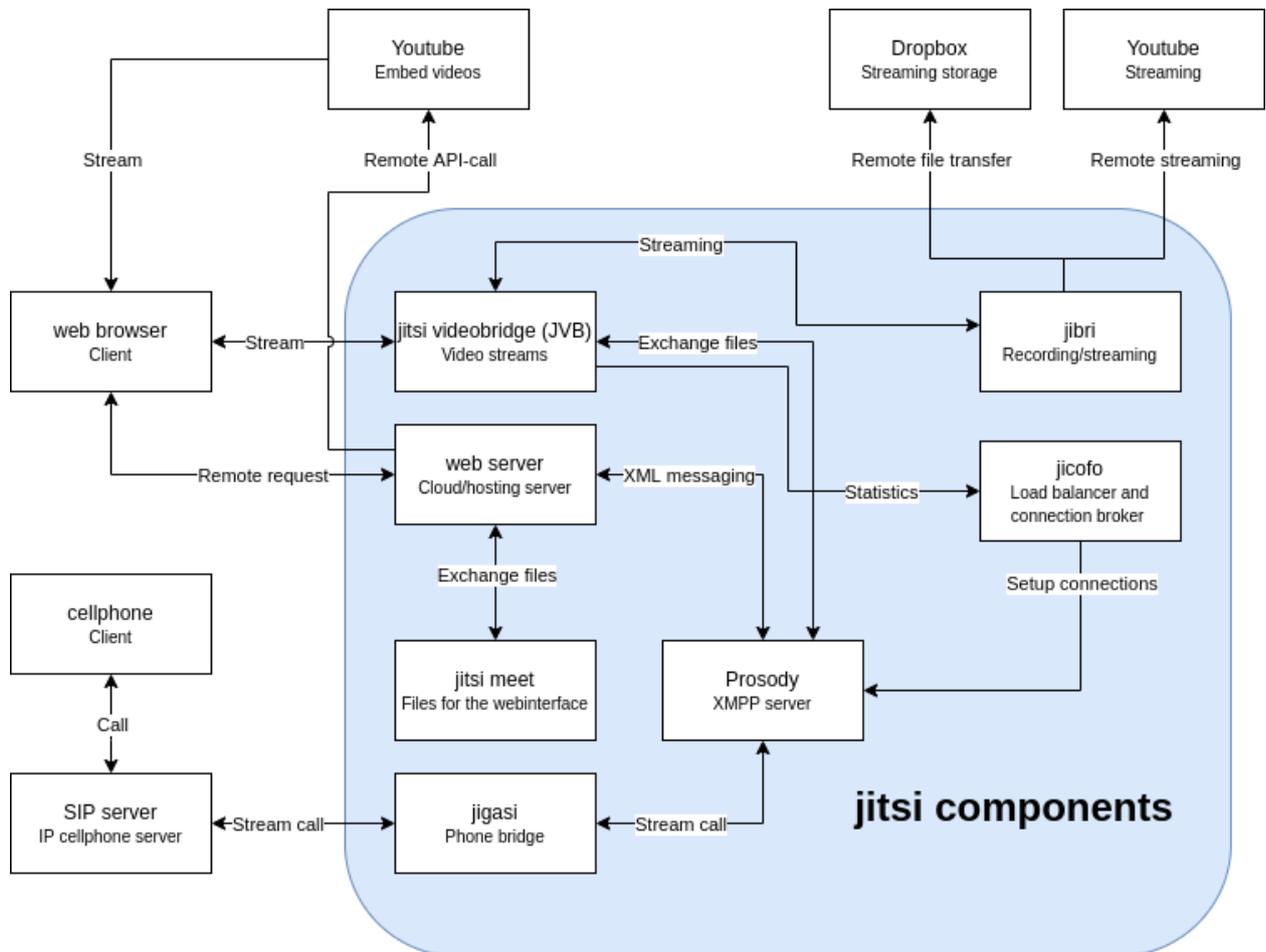


Fig. 4. Arquitectura de Componentes de Jitsi

Fuente: Adaptado de (Architecture · Jitsi Meet Handbook, n.d.)

Las conexiones entre clientes que solicitan una conexión de video o audio se realizan a través de solicitudes remotas y flujos de datos. La segunda categoría de conexiones externas es aquellas a servicios externos que ayudan a almacenar grabaciones, transmitir grabaciones, videos o ayudar a crear reuniones. (Architecture · Jitsi Meet Handbook, n.d.)

1.3. Cloud Computing

El cloud computing o computación en la nube, nace de los términos: cloud y computing. Cloud, o nube, es el símbolo que se usa generalmente para representar Internet. Se establece un concepto de abstracción (sistemas físicos que no se especifican, almacenamiento de datos en ubicaciones desconocidas, acceso ubicuo de los usuarios y administraciones subcontratadas). Y Computing o computación, reúne los conceptos de informática, lógica de coordinación y almacenamiento. (Palos-Sánchez et al., 2017)

Cloud Computing es un servicio que facilita la entrada de información a través de un conjunto de software, hardware, almacenamiento, servicios e interfaces que mantienen en marcha a una gran cantidad de actores o participantes. La computación en la nube tiene grupos de interés clave, que son: proveedores de aplicaciones que brindan la tecnología, infraestructura, plataforma e información respectivas; los desarrolladores de servicios en la nube a menudo son socios de proveedores; los líderes empresariales impulsan la implementación de servicios en la nube en sus organizaciones; y finalmente a los usuarios finales que utilizan servicios en la nube, gratuitos o de pago. (Castro Querembas, 2020)

La computación en la nube se desarrolló para permitir el acceso de red a pedido, ubicuo y rentable a un grupo compartido de recursos informáticos configurables, como una red de servidores, almacenamiento, servicios, etc. Los servicios y aplicaciones, que se pueden aprovisionar y liberar rápidamente, representan un esfuerzo administrativo mínimo. o la interacción de un proveedor de servicios. (Palos-Sánchez et al., 2017)

1.3.1. Capas del Cloud Computing

El Cloud Computing se puede dividir en tres niveles en función de los servicios que actualmente están ofreciendo las empresas, en el futuro pueden surgir otras capas o clasificaciones.

- Infraestructura como servicio

Se puede decir que es la parte física de la nube. En lugar de que los clientes tengan sus equipos en su propio lugar de trabajo, utilizan un alquiler de servicio y cancelan a un

proveedor para que éste sea quien tenga todo ese equipamiento, ya sean discos duros o equipamiento de redes y se encargue de toda la mantención y optimización de dicho equipamiento, el ejemplo de empresa que ofrece este tipo de servicio es Amazon Web Service con su servicio EC2 (Amazon Elastic Compute Cloud), es un “servicio web que proporciona tamaño variable de capacidad de cómputo en la nube y, está diseñado para hacer escala informática web más fácil para los desarrolladores”.(Yagual et al., 2018a)

- Plataforma como servicio

Se encuentra muy ligada a la capa software como servicio, porque es la plataforma en donde se envuelve el software que pone a disposición el proveedor y es el medio de virtualización para el hardware que el cliente arrienda. La empresa que más servicios enfocados a esta capa es Google App Engine, el cual permite desarrollar, compartir y alojar aplicaciones Web de terceros en su vasta infraestructura.(Yagual et al., 2018b)

- Software como servicio

El proveedor de servicio pone a disposición de los clientes su propio software, creando así un ahorro para el cliente en la búsqueda de licencias, manteniendo actualizado siempre el sistema que necesita el client.(Yagual et al., 2018b) En otro campo de la investigación se encontró que existen tres tipos de nubes que se utilizan en el mundo empresarial como le explica en la **¡Error! No se encuentra el origen de la referencia.:**

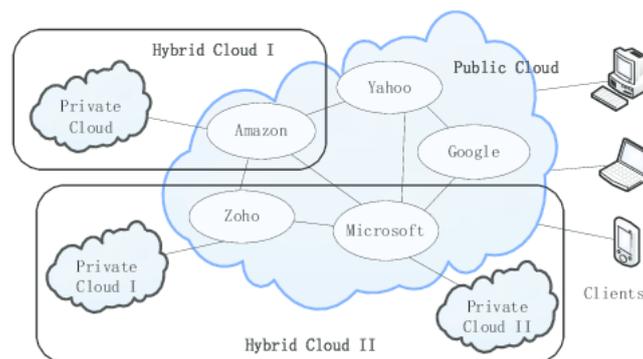


Fig. 5. Tipos de cloud

Fuente: Adaptado de (Yagual et al., 2018b)

1.3.2. Tipos de Cloud Computing

Según Yagua et al. (2018) da una breve explicación de los tipos de Cloud Computing presentados en la Fig. 5:

- Privadas

Las nubes privadas son manejadas por un solo cliente que controla qué aplicaciones debe correr y dónde, son propietarios del servidor, red y disco y que pueden decidir qué usuarios están autorizados a utilizar la infraestructura.

- Públicas

Las nubes públicas se manejan por terceras partes, y los trabajos de muchos clientes de diferentes pueden ser mezclados en los servidores, los sistemas de almacenamiento y otras infraestructuras de la nube. Es importante indicar que los usuarios finales no conocen qué trabajos de otros clientes pueden estar corriendo en el mismo servidor, red, discos como los suyos.

- Híbridas

Las nubes híbridas son las más recomendadas para las empresas pequeñas, combinan los modelos de privada y pública, el propietario es dueño de una parte y comparte otras, actualmente las empresas perciben atracción por la nube híbrida por la facilidad de acoplamiento que tiene a las necesidades de los clientes.

1.4. Bases de Datos no SQL

Una base de datos correctamente diseñado a información exacta y actualizada. Puesto que un diseño correcto es esencial para lograr los objetivos fijados para la base de datos, es lógico emplear el tiempo que sea necesario en aprender los principios de un buen diseño. (Aguirre & Kevin, 2021)

1.4.1. Bases de datos NoSQL

En la actualidad los sistemas web y aplicaciones han cambiado drásticamente en el mundo del internet de las cosas, y debido a ello en los últimos quince años se han realizado cambios en las empresas ya que han tenido la obligación de automatizar todos sus procesos y reestructurar el manejo de su información e infraestructura tecnológica, es por ello que en la última década se ha venido mencionando lo que es Big Data, Big Users y Cloud Computing , dichas tecnologías controlan grandes cantidades información y por esta razón las mismas están teniendo como opción trasladar toda su información hacia las llamadas bases de datos NoSQL ya que ellas poseen herramientas y estructuras adecuadas para simplificar operaciones a gran escala. A demás se puede enfatizar que muchos desarrolladores opinan que es factible utilizar el tipo de modelo de datos que es sin esquema ya que es óptimo para controlar la gran variedad de datos que se procesan hoy en día dentro de las empresas.(Esquivel Vega et al., 2019)

De acuerdo con su estructura y modo de operación con la información las bases de datos NoSQL no cumplen en su totalidad con las características de atomicidad, consistencia, aislamiento y durabilidad (ACID) y debido a ello escalan horizontalmente, por lo que proporcionan de esta manera eficiencia para ser utilizadas en entornos distribuidos.(Esquivel Vega et al., 2019)

1.4.1.1. Beneficios de las Bases de Datos NoSQL

Al contar con una estructura específica promueven beneficios importantes que hacen que se cumplan ciertas características apropiadas para ayudar a los sistemas modernos con el almacenamiento de datos estructurados y no estructurados, de esta manera dichos beneficios pueden ser los siguientes.(Cachimuel, 2020)

- Escalabilidad

Orientado hacia el diseño escalar de clústeres distribuidos donde los nodos se crean de acuerdo con la disponibilidad en que los usuarios necesiten utilizar el almacenamiento en la base de datos, permitiendo de esta manera que se elimine los costos y la complejidad de la fragmentación manual.

- Alto Rendimiento

Optimizado mediante el uso de modelos de datos que se enfocan al control de estructuras documentales, clave-valor y gráficos los cuales mediante patrones dan un acceso confiable y un rendimiento proporcional hacia la información que se esté almacenando.

- Control de datos con flexibilidad

Ofrece libertad para los administradores de bases de datos en el ámbito de velocidad y flexibilidad en el cambio de esquemas y consultas.

- Desarrollo con agilidad

Da soluciones a situaciones imprevistas ya que las bases de datos NoSQL satisfacen de una manera rápida y ágil a las consultas efectuadas para obtener información en grandes cantidades.

1.4.1.2. Riesgos del Uso de Bases de datos NoSQL

Según Cachimuel Bryan (2020) manifiesta que las bases de datos NoSQL presentan fallas en la consistencia que radica en trabajar en donde se plantea diferentes problemas que se han registrado en la seguridad de los datos se dará a conocer los tipos de consistencia que presentan las bases de datos NoSQL.

1.4.1.3. MongoDB

Es una base de datos documental de código abierto y líder en bases de datos NoSQL. MongoDB está escrito en c++. Ofrece alta disponibilidad, escalabilidad y particionamiento a costa de consistencia y soporte transaccional. En términos prácticos, esto significa que, en lugar de tablas y filas, MongoDB utiliza documentos para hacerla flexible, escalable y rápida.(MongoDB: The Application Data Platform | MongoDB, n.d.)

Según Cachimuel Bryan (2020) existe ciertas características sobre mogoDB:

- Permite a los usuarios manipular la información almacenada en forma de documentos con un formato de tipo BSON
- Para la creación de las bases de datos no es necesario la definición de un esquema, lo que permite a los desarrolladores crear sistemas con alta disponibilidad y flexibles a modificaciones que se requieran realizar en la estructura de los documentos.
- Ofrece diferentes tipos de prestaciones a nivel de implementación como es la definición de documentos anidados o referenciados o también el uso de comandos especializados como el operador lookup el cual sustituye las funciones del operador de juntura entre documentos.

Las operaciones con las cuales se puede trabajar proporcionan a los usuarios la ejecución de múltiples consultas en documentos, además de ofrecer la facilidad de manipulación de estos, como es la inserción de documentación mediante tuberías de agregación y mapeos de reducción.(Cachimuel, 2020)

1.5. Metodología de Desarrollo de Software: Extreme Programming (XP)

Extreme Programming (XP) es una metodología ágil para desarrollar y crear software, XP surgió como una metodología que enfatiza la simplicidad y agilidad, las metodologías de desarrollo tradicionales en comparación con XP tienden a volverse ineficientes y pesadas para el desarrollo de equipos, es importante mencionar que XP se enfoca sus esfuerzos para satisfacer eficazmente las necesidades del cliente con una visión clara de Los cuatro aspectos fundamentales de cualquier proyecto de software son: costo, tiempo, calidad y alcance.(Castro Querembas, 2020)

- Consistencia en las escrituras

Se presenta cuando existe un problema de escritura-escritura en donde varios usuarios realizan el proceso de actualización de un mismo dato al mismo tiempo, es decir que al realizar la actualización por parte de los usuarios las órdenes llegan al servidor y la serializan y vuelven a enviar la respuesta de una manera ordenada es allí donde radica el problema ya que un usuario recibe el dato actualizado mientras que al otro usuario se le produce una pérdida de actualización debido a que la última escritura que se aplica se sobrescribe sobre la primera. Una solución que se ha planteado es tener bloqueos de escritura en donde si se desea realizar la actualización de un dato este se bloquee

temporalmente hasta que el usuario termine de actualizarlo y después de realizar este proceso el dato dejará de estar bloqueado y estará disponible para ser utilizado por otros usuarios. (Castro Querembas, 2020)

- Consistencia en las lecturas

Se presenta cuando existe un problema de lectura-escritura en donde un usuario realiza la lectura de un dato en medio de la escritura de otro usuario que en ese momento se encuentra leyendo o escribiendo en un dato, es por esto por lo que sea planteado que las bases de datos relacionales apliquen las transacciones en donde se asegura que el usuario solo realice la lectura y luego la escritura o en su defecto la escritura y luego la lectura. Este concepto afirma que las Bases de Datos NoSQL no soportan transacciones lo que es incorrecto ya que las bases de datos orientadas a grafos pueden llegar a soportar cierto tipo de transacciones, como también las Bases de datos orientadas hacia agregados que puede soportar actualizaciones atómicas sobre un agregado. (Castro Querembas, 2020)

- Consistencia de sesión

Puede llegar a ser un problema las ventanas de inconsistencia ya que casi siempre se llega a mantener una carga que se encuentra ejecutándose en un cluster que mantenga la carga de peticiones a los diferentes nodos donde sea solicitada una ventana, para controlar esta clase de situaciones es necesario implementar un tipo de consistencia de tipo leer lo que se escribe, lo que implica aplicar el concepto de actualización en donde se pueda verificar que solamente se pueda ver la actualización ejecutada por el usuario. (Aguirre & Kevin, 2021)

1.5.1. Objetivos de Extreme Programming

El objetivo de todo desarrollo de software es entregar un producto de calidad, eficiente y sobre todo funcional, encuadrada en esa visión XP posee varios objetivos claros para conseguir un resultado integral favorable (Coloma et al., 2017), algunos de estos se mencionan a continuación.

- Definir al cliente y sus requisitos de manera efectiva.

- Decidir las mejores prácticas de Ingeniería de Software de acuerdo con el proyecto específico.
- Maximizar la productividad de los desarrolladores.
- Motivar al equipo de desarrollo inculcar habilidades como una alta capacidad de aprender.

1.5.2. Etapas y ciclo de Extreme Programming

Según Castro Querembas (2020) Extreme Programming se desarrolla en ciclos y para su desenvolvimiento presenta varias etapas que se presentan a continuación:

- Exploración

Se exploran las diferentes posibilidades de arquitecturas para el proyecto, el cliente realiza a breve rasgos el panorama del proyecto y realiza historias de usuario, mientras los desarrolladores se familiarizan con las tecnologías y buenas prácticas necesarias.

- Planificación de la entrega

El cliente establece la prioridad de cada historia de usuario, en base a esa prioridad se estima el tiempo y esfuerzo necesario para cumplir con cada una de las historias.

- Iteraciones

El plan de entrega se compone por iteraciones que suelen variar entre dos y tres semanas, por lo general en la primera iteración se establece la arquitectura del proyecto.

- Producción

Se trata de llevar el desarrollo al entorno del cliente, se suelen realizar varias pruebas para verificar la calidad de la pieza de software antes de realizarlo.

- Mantenimiento

Una vez se ha implementado la primera versión del proyecto en producción se realizan nuevas iteraciones incluyendo algunos cambios y reformas pedidas por el cliente estas reformas suelen ser incluidas para dar soporte al cliente.

- Muerte del Proyecto

Se concluye el proyecto una vez cumplidas las historias de usuario realizadas por el cliente, se realiza la documentación final del proyecto, esta etapa implica que la arquitectura del proyecto no tendrá más cambios.

1.5.3. Roles de la Metodología Extreme Programming

Extreme Programming cuenta para su desenvolvimiento con varios roles según Castro Querembas (2020) en su trabajo de titulación detalla los siguientes breves rasgos de cada uno.

- Programador

Es alguien que produce el código del sistema y escribe pruebas unitarias.

- Cliente

Realiza las historias de usuario y pruebas funcionales, con la finalidad de comprobar su validez en la implementación.

- Tester

Asesora al cliente en la redacción de pruebas funcionales, además es quien regularmente ejecuta las pruebas y expone los resultados al equipo.

- Tracker

Proporciona realimentación al equipo en el proceso de desarrollo ágil XP, estima el grado de acierto en las estimaciones realizadas y el tiempo real que se le ha dedicado al proyecto.

- Entrenador

El encargado y responsable del proceso en ambiente global, conoce a profundidad el proceso XP y provee al equipo las guías de la metodología ágil.

- Consultor

Posee un conocimiento específico en algún tópico fundamental para el proyecto.

- Gestor

Se encarga de vigilar que el equipo trabaje en condiciones adecuadas, además es el vínculo entre los programadores y el cliente.

1.6. Backend y Frontend

Según Ortega Checa (2020) un desarrollador backend es quien trabaja del lado del servidor, utilizando lenguajes tales como JavaScript, C#, Python, etc; interactuando con bases de datos, verificando sesiones de usuario y montando una página en el servidor.

1.6.1. Node JS como servidor (Backend)

Node.js es un entorno de ejecución de JavaScript de código abierto y multiplataforma. Este ejecuta el motor JavaScript V8, el núcleo de Google Chrome, fuera del navegador. Esto permite que Node.js sea muy eficaz. (Introduction to Node.js, n.d.)

Una aplicación Node.js se ejecuta en un solo proceso, sin crear un nuevo hilo para cada solicitud. Node.js proporciona un conjunto de primitivas de E / S asíncronas en su biblioteca estándar que evitan que el código JavaScript se bloquee y, en general, las bibliotecas en Node.js se escriben utilizando paradigmas sin bloqueo, lo que hace que el comportamiento de bloqueo sea la excepción y no la norma. (Introduction to Node.js, n.d.)

Cuando Node.js realiza una operación de E / S, como leer desde la red, acceder a una base de datos o al sistema de archivos, en lugar de bloquear el hilo y desperdiciar ciclos de CPU esperando, Node.js reanudará las operaciones cuando vuelva la respuesta. Esto permite que Node.js maneje miles de conexiones simultáneas con un solo servidor sin

introducir la carga de administrar la concurrencia de subprocessos, lo que podría ser una fuente importante de errores. (Introduction to Node.Js, n.d.)

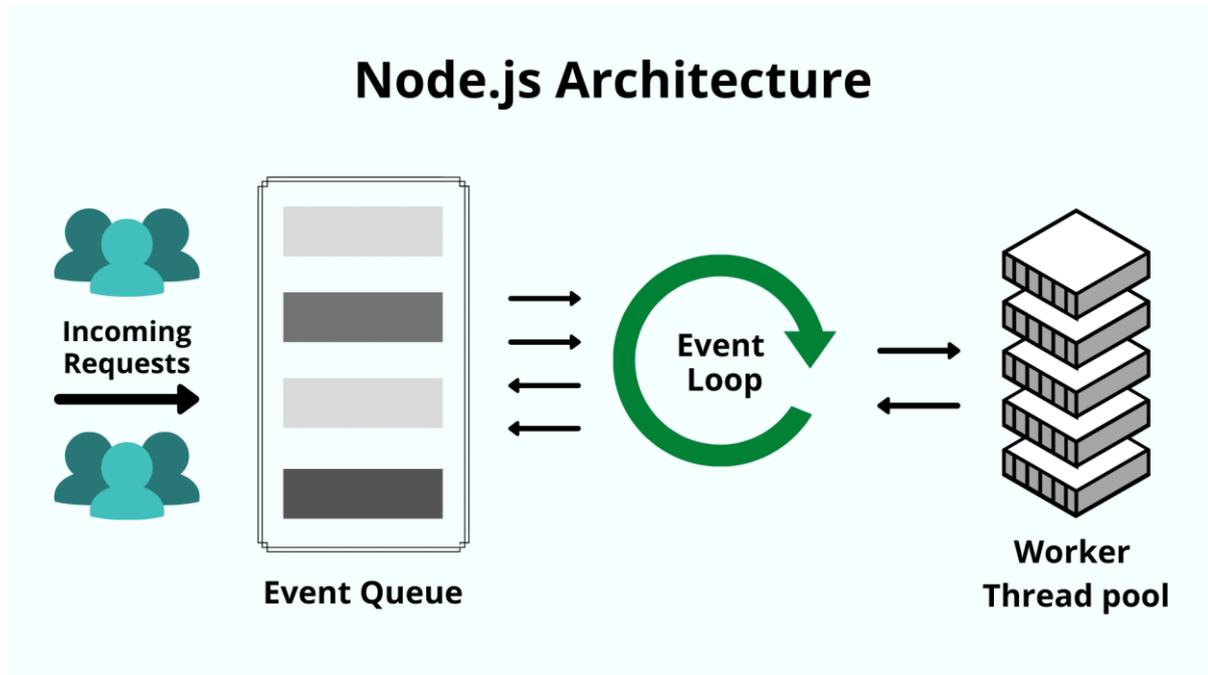


Fig. 6. Arquitectura node js

Fuente: Adaptade de (*Qué Es Node.Js y Por Qué Deberías Usarlo*, n.d.)

1.6.2. React JS para creación de interfaces (Frontend)

React es una librería que ayuda a crear interfaces de usuario interactivas de manera sencilla y rápida. Diseña vistas simples para cada estado en la aplicación, y React se encarga de actualizar y renderizar de manera eficiente los componentes correctos cuando los datos cambien, las vistas como componentes hacen que el código sea más predecible, por lo tanto, fácil de depurar.(React – Interfaces de Usuario, n.d.)

React crea componentes encapsulados que manejen su propio estado, y conviértelos en interfaces de usuario complejas. Ya que la lógica de los componentes está escrita en JavaScript y no en plantillas, puedes pasar datos de forma sencilla a través de tu aplicación y mantener el estado fuera del DOM. (React – Interfaces de Usuario, n.d.)

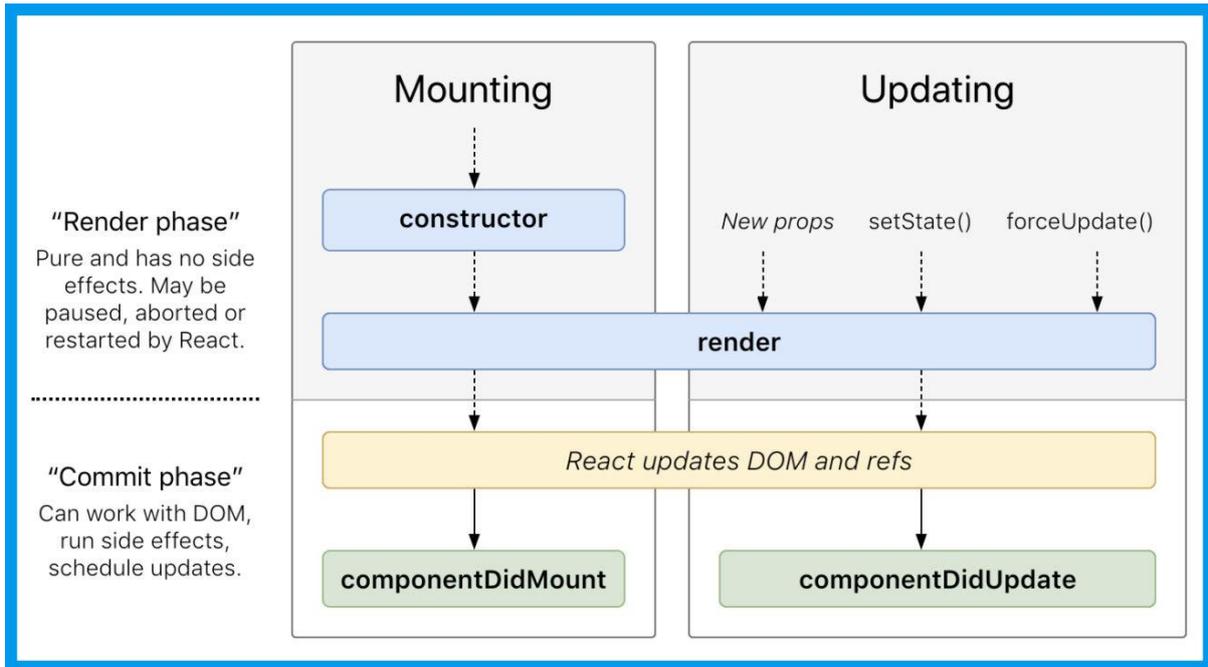


Fig. 7. Ciclo de vida de un componente

Fuente: Adaptade de (React – Interfaces de Usuario, n.d.)

CAPÍTULO 2

2. Desarrollo

2.1. Planificación

El software se desarrolla teniendo como objetivo el crear, editar, eliminar y editar videoconferencias con la librería de software libre jitsi, la construcción de un calendario para la creación de reuniones tipo webinar y el envío posterior de la información de dichas reuniones al correo de las personas que estén interesadas en el evento, para ello fue necesario la creación de API's y creación de interfaces que consuman peticiones http, acorde a lo solicitado por la empresa IERec.

A continuación, se detallará en la TABLA 1 los partes que conforman el módulo:

TABLA 1.
Partes del módulo de videoconferencias

Sistema	Descripción
Calendario	<p>Sistemas web que estará conformado por la creación de un API Rest para las diferentes acciones del calendario, donde posteriormente se consumirán en el frontend donde se realizara las siguientes funciones:</p> <ul style="list-style-type: none">• Crear, consultar, editar y eliminar las videoconferencias.• Publicación de las videoconferencias.• Envío de la información de las videoconferencias mediante correo.
Videoconferencias	<p>Sistema de videoconferencias mediante la ayuda de la librería jitsi de código abierto donde se realiza las videoconferencias este sistema realizara las siguientes funciones principales:</p> <ul style="list-style-type: none">• Creación de videoconferencias.• Pagina prejoin donde se podrá ver la configuración del micrófono, cámara, ingreso de nombre y fondos virtuales.• El moderador podrá silenciar, expulsar participantes, poner contraseña a las videoconferencias y silenciar sonidos de reacción.

2.1.1. Metodología de Desarrollo

Para el desarrollo del módulo se implementó la metodología Extreme Programming (XP) con el que a continuación se da a conocer los detalles de cada una de sus etapas, algunas de las características de esta metodología fueran a adaptadas acorde a las posibilidades del proyecto.

- Roles de desarrollo:

Se definen los roles de desarrollo en la TABLA 2 que se consideran importantes para el avance de las actividades del proyecto que se evidencia en la TABLA 1:

TABLA 2.
Roles Extreme Programming

Rol	Integrante del Equipo
Cliente	Ing. Guillermo Pérez
Consultor	
Entrenador Gestor	Ing. Cosme MacArthur Ortega Bustamante
Programador	Pedro Jhoel Malte Andrade
Tester	

2.1.2. Levantamiento de requerimientos

El levantamiento de requerimientos se realizó en varias reuniones de trabajo, donde el cliente dio a conocer los puntos a desarrollar del módulo de videoconferencias, como puntos importantes se mencionan los siguientes:

- El módulo de videoconferencias es un componente de un entorno virtual de aprendizaje integrado (evai).
- El módulo de videoconferencias contara con un calendario donde se gestionarán las videoconferencias y un sitio donde se realizarán las videoconferencias.
- El módulo de videoconferencias deberá ser integrado en el entorno virtual de aprendizaje integrado (evai).
- Se realizarán pruebas de estrés para verificar la eficiencia de las videoconferencias.

- Cuando el sistema esté desplegado en producción deberá contar con certificados de seguridad SSL.

2.1.3. Casos de Uso

De acuerdo con los requisitos del software se realizó una secuencia de acciones a través de los casos de uso proporcionando una estructura para expresar los requisitos funcionales donde se vería que una secuencia de acciones. Para este módulo se cuenta con 2 actores administrador y usuario.

- Administrador

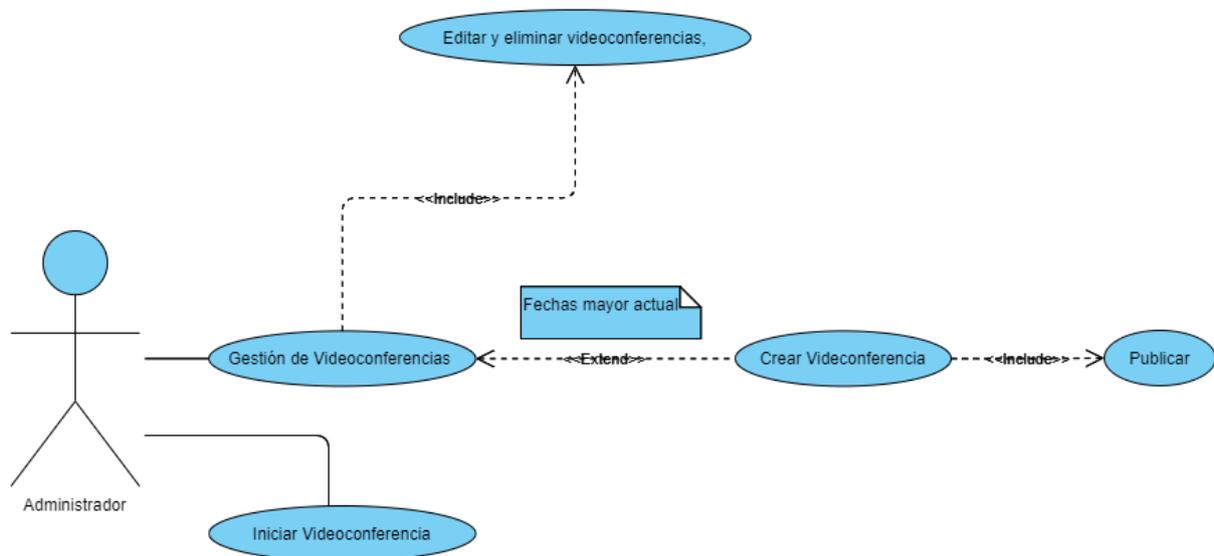


Fig. 8. Caso de uso administrador

Fuente: Propia

En la TABLA 3 se muestra las características del actor administrador.

TABLA 3.
Descripción actor administrador

Nombre	Descripción
Identificador	Administrador
Actor	Administrador
Propósito	Gestión de videoconferencias

Visión general

El administrador crea, edita, modifica y elimina las videoconferencias en un calendario además de iniciar las mismas.

- Usuario

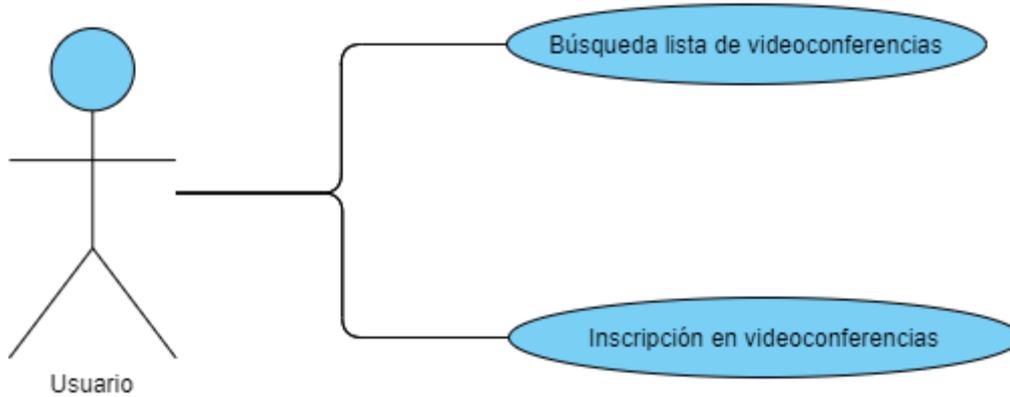


Fig. 9. Caso de uso usuario

Fuente: Propia

En la TABLA 4 se muestra las características del actor usuario.

TABLA 4.
Descripción actor usuario

Nombre	Descripción
Identificador	Usuario
Actor	Usuario
Propósito	Buscar y registrarse en una videoconferencia
Visión general	El usuario podrá buscar en una lista de videoconferencias donde deberá ingresar su nombre y correo en las cuales podrá registrarse y la información de la videoconferencia será enviada al correo proporcionado.

Fuente: Propia

2.1.4. Historias de Usuario

Se hace un listado de historias en base a las reuniones con el usuario y levantamiento de requerimientos. En la TABLA 5 se especifica un listado de las historias de usuario donde se detalla el nombre de la historia, prioridad y riesgo con los valores alto, medio y bajo también se especifica el número de iteración que ayuda a saber el orden en el que se desarrollara.

TABLA 5.
Listado historias de usuario

Nro.	Nombre	Prioridad	Riesgo	Iteración
H1	Preparación del ambiente de desarrollo uso de Jitsi.	Alta	Medio	1
H2	Configuración de las variables de desarrollo.	Alta	Medio	1
H3	Creación de componente prejoin de las videoconferencias.	Medio	Bajo	1
H4	Creación del baúl de certificados.	Alta	Bajo	2
H5	Configuración de puertos seguros.	Alta	Bajo	2
H6	Gestión de videoconferencias.	Alta	Bajo	2
H7	Inscripción videoconferencia.	Alta	Medio	3
H8	Integración al sistema EVAL.	Alta	Medio	3
H9	Compra Droplet Digital Ocean	Alta	Baja	3

Fuente: Propia

2.1.5. Iteraciones

- Primera iteración.

TABLA 6.
Historia de usuario H1

HISTORIA DE USUARIO	
Número: 1	Sistema: Videoconferencias
Nombre: Preparación del ambiente de desarrollo para el uso de Jitsi.	
Prioridad: Alta	Riesgo: Medio
Descripción: Descarga, preparación y configuración del ambiente de desarrollo para el uso adecuado de la librería jitsi.	
Observaciones: Ninguna.	

TABLA 7.
Historia de usuario H2

HISTORIA DE USUARIO	
Número: 2	Sistema: Videoconferencias
Nombre: Configuración de las variables de desarrollo.	
Prioridad: Alta	Riesgo: Medio
Descripción: Uso de la librería de jitisi para configurar los componentes a usar en la interfaz de la videoconferencia.	
Observaciones: Ninguna.	

TABLA 8.
Historia de usuario H3

HISTORIA DE USUARIO	
Número: 3	Sistema: Videoconferencias
Nombre: Creación de componente prejoin de las videoconferencias.	
Prioridad: Media	Riesgo: Bajo
Descripción: El usuario podrá configurar el uso de la cámara, micrófono e ingresar su nombre de usuario antes de ingresar a una conferencia.	
Observaciones: Ninguna.	

- Segunda iteración.

TABLA 9.
Historia de usuario H4

HISTORIA DE USUARIO	
Número: 4	Sistema: Videoconferencias
Nombre: Creación del baúl de certificados.	
Prioridad: Alta	Riesgo: Bajo
Descripción: Se crea un baúl de certificados donde se podrán almacenar certificados ssl para tener conexiones seguras.	
Observaciones: La librería de jitisi no permite la creación de videoconferencias si no contienen certificados de seguridad.	

TABLA 10.
Historia de usuario H5

HISTORIA DE USUARIO	
Número: 5	Sistema: Videoconferencias
Nombre: Configuración de puertos seguros.	
Prioridad: Alta	Riesgo: Bajo
Descripción: Se configura los puertos seguros para su posterior despliegue en un servidor en la nube.	
Observaciones: Se hizo el uso del puerto 80 y 443 ya que sin estos puertos en un servidor bloquea la creación de videollamadas.	

TABLA 11.
Historia de usuario H6

HISTORIA DE USUARIO	
Número: 6	Sistema: Videoconferencias
Nombre: Gestión de videoconferencias.	
Prioridad: Alta	Riesgo: Bajo
Descripción: Las videoconferencias se gestionarán mediante un calendario, estas videoconferencias se podrán crear, editar, eliminar y mostrarán en un landing donde las personas podrán tener más información de estas.	
Observaciones: Solo el administrador podrá crear videoconferencias.	

- Tercera iteración.

TABLA 12.
Historia de usuario H7

HISTORIA DE USUARIO	
Número: 7	Sistema: Calendario
Nombre: Inscripción videoconferencia.	
Prioridad: Alta	Riesgo: Medio

Descripción: Los usuarios podrán ver las conferencias creadas en un landing donde podrán llenar sus datos y se enviara a su correo la información de las videoconferencias para su posterior unión a estas.

Observaciones: La información sobre las videoconferencias solo se enviará a su correo una vez que llene los datos que se pidan en un formulario.

TABLA 13.
Historia de usuario H8

HISTORIA DE USUARIO

Número: 8

Nombre: Integración al sistema EVAI.

Prioridad: Alta

Riesgo: Medio

Descripción: El módulo de videoconferencias será integrado como componente al entorno virtual de aprendizaje integrado (evai).

Observaciones: Ninguna.

TABLA 14.
Historia de usuario H9

HISTORIA DE USUARIO

Número: 9

Nombre: Compra Droplet Digital Ocean.

Prioridad: Alta

Riesgo: Bajo

Descripción: **Comprar** un droplet dependiendo de las especificaciones que se necesite para el despliegue del modulo

Observaciones: El cliente facilitará una cuenta en Digital Ocean donde estará disponible un monto de dinero para la compra del droplet.

2.2. Diseño

2.2.1. Arquitectura del sistema

En la Fig. 10 se detalla el diseño del sistema web su estructura y sus componentes como estará construido desde el inicio al final del despliegue.

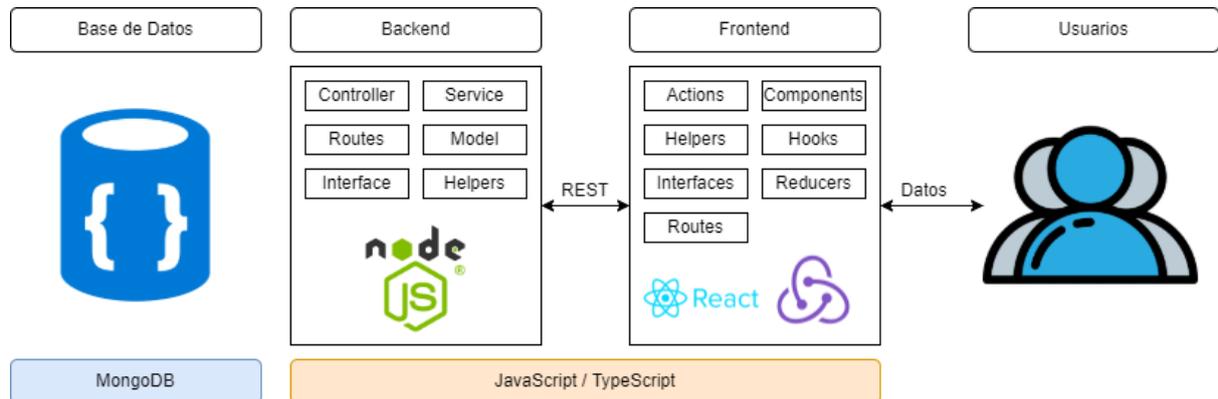


Fig. 10. Arquitectura del sistema

Fuente: Propia

2.2.2. Diagrama de procesos

A continuación, se presenta los diagramas de proceso de las partes del software que se especifica en la TABLA 1.

2.2.2.1. Diagrama de proceso calendario

Este proceso ayuda a la automatización de la gestión de videoconferencias. En la Fig. 11 podemos ver todos los pasos que realiza el proceso de calendario.

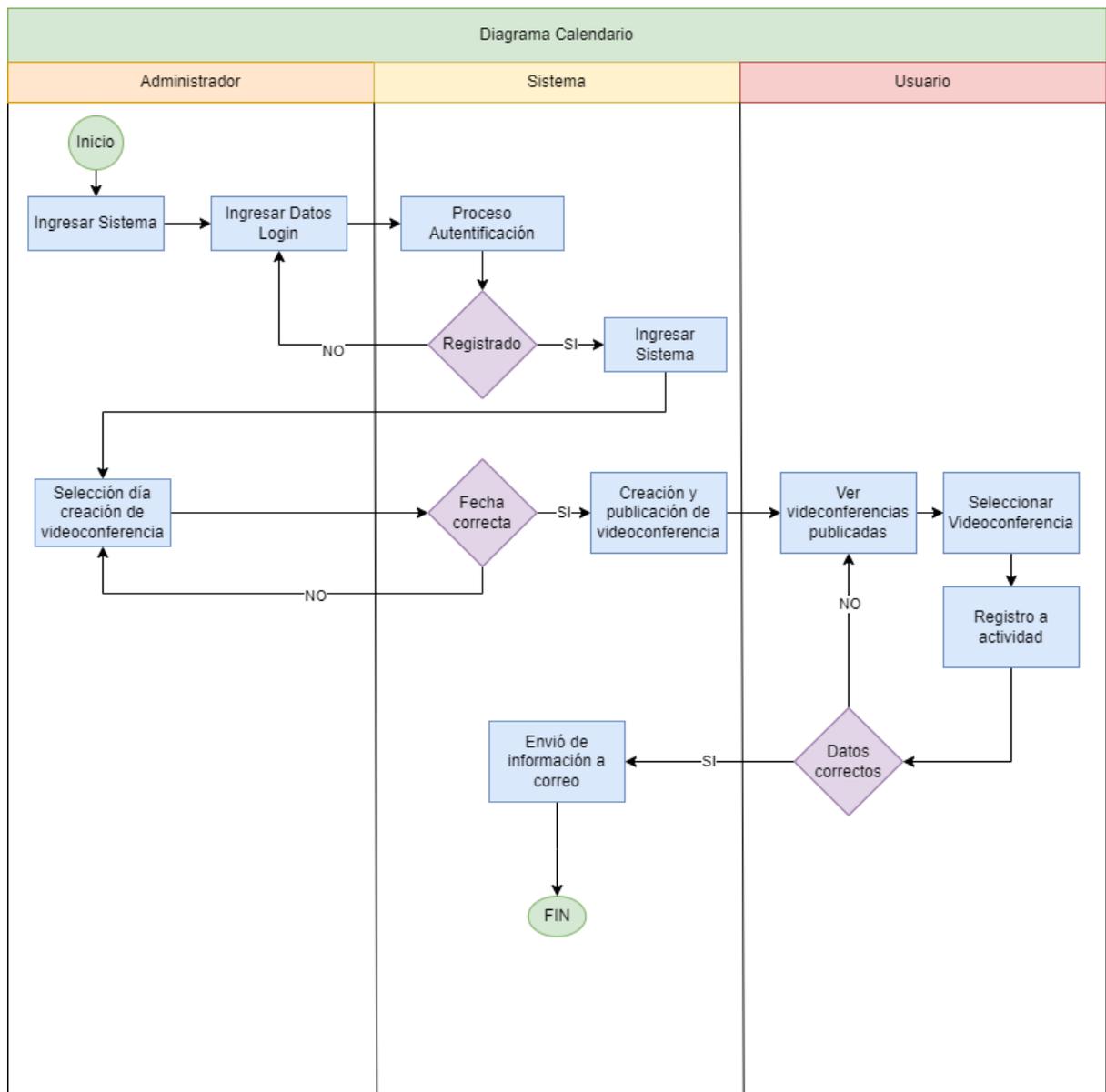


Fig. 11. Diagrama de proceso calendario

Fuente: Propia

2.2.2.2. Diagrama de proceso videoconferencias

Este proceso ayuda a la creación de videoconferencias. En la Fig. 12 podemos ver todos los pasos que realiza el proceso de videoconferencias.

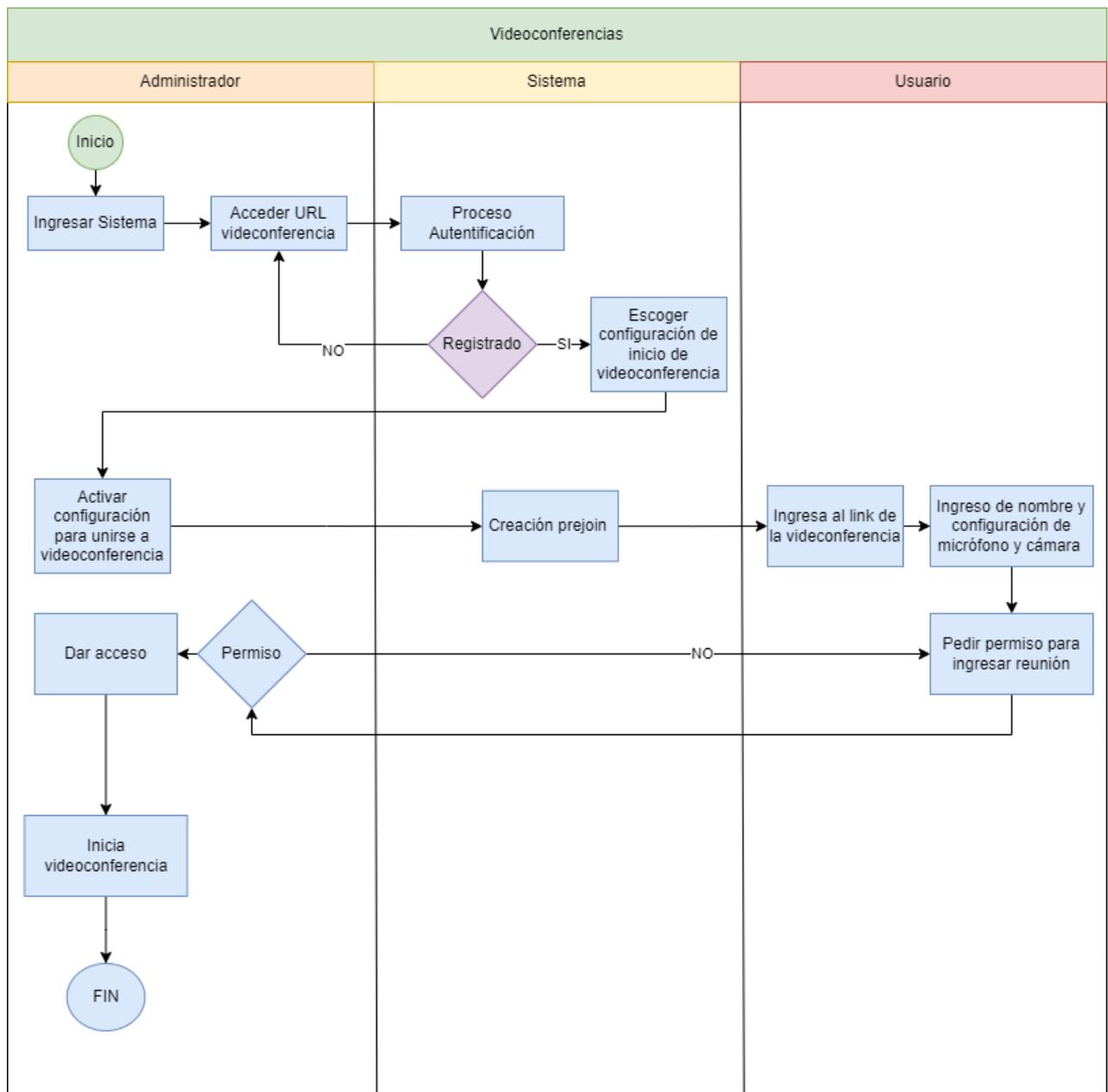


Fig. 12. Diagrama de proceso videoconferencia

Fuente: Propia

2.2.3. Arquitectura base de datos

Se detalla el modelo del base de datos NoSQL en mongoDB que utiliza el sistema web, cabe aclarar que a diferencia de las bases de datos relaciones que cuentan con tablas, columnas, filas o esquemas para organizar y obtener datos, mongoDB no sigue este tipo de organización, sino que guarda una estructura de datos JSON (Giordani et al., 2021), en la se Fig. 13 da un ejemplo del cómo se estructuran los datos en JSON.

```
_id: ObjectId("61d24ee8d3acb360112e3c8b")
nombre: "revision" tesis
link: " "
img: "575502310.jpeg"
fecha: 2022-01-31T22:50:00.000+00:00
detalle: "Esta reunion trate de"
title: "curso angular"
__v: 0
```

Fig. 13. Estructura de datos JSON en mongoDB

Fuente: Propia

A continuación, en la Fig. 14 se muestra el modelo de la base de datos que sirve para el registro de información de la gestión de videoconferencias.

```
const meetSchema = new Schema<i_meeting, Model<i_meeting>, i_meeting>(
  {
    nombre: {
      type: String,
      required: false,
    },
    link: {
      type: String,
      required: false,
    },
    img: {
      type: String,
      required: false,
    },
    fecha: {
      type: Date,
      required: true,
    },
    detalle: {
      type: String,
      required: false,
    },
    title: {
      type: String,
      required: true,
    },
  },
  { collection: 'meeting' }
)

export default model<i_meeting_doc>('Meeting', meetSchema)
```

Fig. 14. Modelo base de datos noSQL meeting

Fuente: Propia

2.3. Codificación

En la codificación del sistema se realizó la configuración de las diferentes herramientas y se desarrolla el sistema web.

2.3.1. Configuración del entorno de desarrollo

En este bloque se realizó todas las instalaciones y configuraciones de las herramientas que se usaron para el desarrollo del sistema en la TABLA 15 se da un breve detalla del software usado.

TABLA 15.
Software necesario

Software	Detalle
Visual Code	IDE escogida para realizar la codificación por su versatilidad y rapidez para escribir código.
Postman	Postman es una aplicación de cliente http que nos permite realizar pruebas API.
MongoDB	Base de datos NoSQL orientado a documentos.
MongoDB Compass	Es una interfaz gráfica que facilita el uso y la gestión de MongoDB
Google Chrome	Navegador web donde se desplegará el sistema para pruebas.
Git	Software para el control de versiones
Docker	Ayuda al despliegue de aplicaciones en contenedores.

2.3.2. Creación de variables de entorno

En este bloque se realizó creación de variables de entorno que permite configurar el software separado del código. Para calendario se usa 2 tipos de variables de entorno development y production en la Fig. 15 se da un ejemplo de cómo son las variables de entorno usadas en el sistema, para las videoconferencias se usa otro tipo de variable de entorno llamado jitsi enviroment.

```
PORT=4000
DB_HOST=localhost
DB_USER=root
DB_PASSWORD=
DB_NAME=Agenda
UserMail=
PasswordMail=
HostMail=xxxx@gmail.com
```

Fig. 15. Ejemplo de variables de entorno

Fuente: Propia

- Development: Son variables de entorno de tipo desarrollador que sirven cuando el sistema es ejecutado en localhost.
- Production: Son variables de entorno de tipo producción que sirven cuando el sistema es desplegado para el consumo del cliente.
- Jitsi enviroment: Variables de entrono jitsi que sirve para configurar el sistema y sirve para desplegar a producción en la Fig. 16 se observa las variables más básicas necesarias para producción.

```
# Configuraciones Basicas

# HTTP port
HTTP_PORT=8000

# HTTPS port
HTTPS_PORT=8443

# Zona horaria del sistema
TZ=America/Guayaquil

# URL pública para el servicio web
#PUBLIC_URL=https://jitsitest.me

# Controle si la función del lobby debe estar habilitada o no
#ENABLE_LOBBY=1

# Mostrar pagina pre incial antes de entrar a meeting
#ENABLE_PREJOIN_PAGE=0

# Mostrar pagina de bienvenida
#ENABLE_WELCOME_PAGE=1

# Mostrar pagina de despedida
#ENABLE_CLOSE_PAGE=0

# Habilitar salas para grupos pequeños
ENABLE_BREAKOUT_ROOMS=1

#
# Configuración de autenticación
#

# Habilitar la autenticación
#ENABLE_AUTH=1

# Habilitar el acceso de invitados
#ENABLE_GUESTS=1
```

Fig. 16. Variables de entorno jitsi

Fuente: Propia

2.3.3. Configuración y ejecución de la base de datos NoSQL de calendario

Para la configuración y ejecución de la base de datos mongoDB se hizo uso de la librería moongoose que sirve para hacer la conexión y gestión de datos desde el código.

- Configuración base de datos mongoDB: en esta configuración se puede ver en la Fig. 17, se hace uso de las variables de entorno de la Fig. 15 que dependiendo de donde se ejecute el proyecto tomará las variables de development o production y hará la conexión mediante la librería de mongoose.

```
import { ConnectOptions } from 'mongoose'
import {
  NODE_ENV,
  DB_USER,
  DB_PASSWORD,
  DB_HOST,
  DB_NAME,
  DB_PORT,
} from '@config/config'

export const MongoOpcion: ConnectOptions = {
  autoIndex: false,
  maxPoolSize: 10,
  socketTimeoutMS: 45000,
  serverSelectionTimeoutMS: 5000,
}

export const Url = (): string => {
  let rest = `mongodb://${DB_HOST}:${DB_PORT}/${DB_NAME}`
  if (NODE_ENV === 'production') {
    rest = `mongodb+srv://${DB_USER}:${DB_PASSWORD}@${DB_HOST}/${DB_NAME}`
  }

  return rest
}
```

Fig. 17. Configuración base de datos mongoDB

Fuente: Propia

- Ejecución base de datos mongoDB: Cuando la configuración es correcta se procede a realizar la conexión cuando se ejecuta el proyecto la base de datos solo será creada una vez que se realice una inserción datos ejecutando el modelo de la Fig. 18.

```
mongoose.connect(
  process.env.URLDB,
  { useNewUrlParser: true, useUnifiedTopology: true },
  (err, res) => {
    if (err) throw err
    console.log('Base de datos conectada')
  }
)
```

Fig. 18. Ejecución y conexión base de datos mongoDB

Fuente: Propia

2.3.4. Creación backend

El backend es el sitio donde se desarrolla y se crea las configuraciones, servicios, rutas, interfaces y los modelos de la base de datos. El proyecto que se va a ejecutar en un entorno de desarrollo nodeJS.

Para crear el proyecto de calendario con nodeJS se realiza con el `"npm init -y"` en la Fig. 19 se muestra la creación del proyecto.

```
PS C:\Users\jowel\Desktop\tesis-Calendario> npm init -y
Wrote to C:\Users\jowel\Desktop\tesis-Calendario\package.json:

{
  "name": "tesis-calendario",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Fig. 19. Ejecución comando creación backend

Fuente: Propia

En la Fig. 20 se pueda observar la creación de diferentes carpetas de la estructura de las carpetas del software donde se comienza a implementar las diferentes clases y métodos.

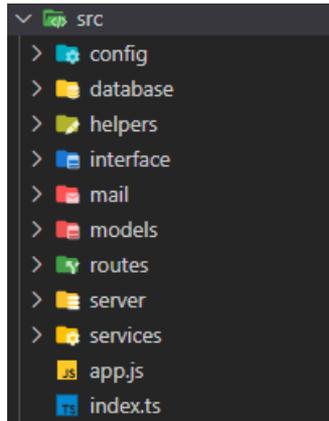


Fig. 20. Estructura de las carpetas backend

Fuente: Propia

En cambio, para la creación del proyecto videoconferencia se hizo descarga del repositorio oficial de jitsi en github para la posterior construcción del backend con los siguientes comandos:

- Descarga del repositorio

Para descargar el repositorio oficial de jitsi se aplicó el comando de la Fig. 21.

```
jowel@DESKTOP-0D6GGB6 MINGW64 ~/Desktop
$ git clone https://github.com/jitsi/jitsi-meet.git
Cloning into 'jitsi-meet'...
remote: Enumerating objects: 104895, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (9/9), done.
Receiving objects: 20% (21345/104895), 12.49 MiB | 1.95 MiB/s
```

Fig. 21. Repositorio oficial jitsi

Fuente: Propia

- Construcción backend

Para construir el proyecto debemos instalar todos los paquetes con el comando de la Fig. 22.

```
jowel@DESKTOP-0D6GGB6 MINGW64 ~/Desktop/jitsi-meet (master)
$ npm install
npm WARN deprecated @hapi/topo@3.1.6: This version has been deprecated and is no longer supported or maintained
npm WARN deprecated @hapi/bourne@1.3.2: This version has been deprecated and is no longer supported or maintained
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated sane@4.1.0: some dependency vulnerabilities fixed, support for node < 10 dropped, and newer ECMAScript syntax/features added
npm WARN deprecated @hapi/address@2.1.4: Moved to 'npm install @sideway/address'
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParams API instead.
npm WARN deprecated uuid@3.1.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
```

Fig. 22. Instalación paquetes jitsi

Fuente: Propia

2.3.5. Creación de rutas en el backend

En este bloque se crean las rutas con la librería Express que es una infraestructura web de direccionamiento una aplicación es fundamentalmente una serie de llamadas a funciones de middleware (*Express*, n.d.). Esta librería proporciona un routing API con funciones:

- use()
- params()
- route()

También podemos consumir peticiones http como GET, POST, PUT y DELETE, en la Fig. 23 se muestra las rutas más básicas usadas.

```

export default class MeetingRoute {
  static Routes() {
    const router = Router()
    router.route('/allMeetings').get(getAllMeetings)

    router.route('/activeMeetings').get(getActiveMeetings)

    router.route('/createMeeting').post(upload.single('img'), createMeeting)

    router.route('/deleteMeeting/:id').delete(deleteMeeting)

    router
      .route('/editeMeeting/:id')
      .put(upload.single('img'), editeMeeting)

    router.route('/enviarEmails/:id').post(enviarEmail)

    return router
  }
}

```

Fig. 23. Creación de peticiones http

Fuente: Propia

2.3.6. Creación de servicios en el backend

En este bloque se realiza la creación de los servicios los cuales nos ayudan a procesar la información que nos viene por las peticiones http, mediante los servicios se puede interactuar con la base de datos. Estos servicios nos ayudan a crear, editar y eliminar datos de la base de datos a continuación se presenta algunos servicios básicos usados.

- getAllMeetings [GET]
- createMeeting [POST]
- deleteMeeting [DELETE]
- editeMeeting [PUT]

En la Fig. 24 se muestra la estructura de uno de los servicios básicos más usadas en lo largo de la creación del código para consumo de peticiones http.

```

export const getAllMeetings = (req: Request, res: Response) => {
  meetingModel.find().exec((err: any, data: any) => {
    if (err) {
      return res.status(400).json({
        ok: false,
        message: 'No se pudieron encontrar las meetings',
        err,
      })
    }

    res.status(200).json(data)
  })
}

```

Fig. 24. Método obtención de todas las videoconferencias

Fuente: Propia

2.3.7. Configuración de Envío de correos en el backend

En este bloque se realiza la configuración para el envío de la información de las videoconferencias. Para el envío de correos se usó la librería nodemailer que es una librería que ayuda al envío de correos electrónicos. Para poder configurar nodemailer se hace el uso de 2 metodos que se presentan a continuación:

- Encabezados

Método de configuración de las variables que se pueden observar en la Fig. 25 para el envío del correo.

```

import { USER_MAIL, PASSWORD_MAIL, HOST_MAIL } from '@config/config'

export const transportConfig = {
  host: HOST_MAIL,
  port: 465,
  secure: true,
  auth: {
    user: USER_MAIL,
    pass: PASSWORD_MAIL,
  },
}

```

Fig. 25. Información correo

Fuente: Propia

- Transport

Método de envío que se crea a partir de una promesa para asegurar que la información llegue de manera correcta en la Fig. 26 se puede observar parte del código. Transporter hace uso de la configuración de los encabezados que se ve en la Fig. 25.

```
public sendMail(emailConfig: SendMailOptions): Promise<boolean> {
  return new Promise((res, rej) => {
    return this.Transporter.sendMail(emailConfig, (err: any) => {
      if (err) return rej(false)
      return res(true)
    })
  })
}
```

Fig. 26. Método envío de correo

Fuente: Propia

2.3.8. Creación frontend

El frontend es el sitio donde se desarrolla los diseños y se configura el comportamiento de la ejecución en el navegador y además también se encarga de dar interactividad a los usuarios.

Para crear el proyecto con react se realiza con el comando "npx create-react-app proyecto-tesis" en la Fig. 27 se muestra la ejecución del comando.

```
jowel@DESKTOP-0D6GGB6 MINGW64 ~/Desktop
$ npx create-react-app proyecto-tesis
npm WARN exec The following package was not found and will be installed: create-react-app
npm WARN deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security updates. Please upgrade asap.

Creating a new React app in C:\Users\jowel\Desktop\proyecto-tesis.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
```

Fig. 27. Ejecución comando creación frontend

Fuente: Propia

En la Fig. 28 se pueda observar la creación de diferentes carpetas de la estructura del software donde se comienza a implementar las diferentes clases y métodos.

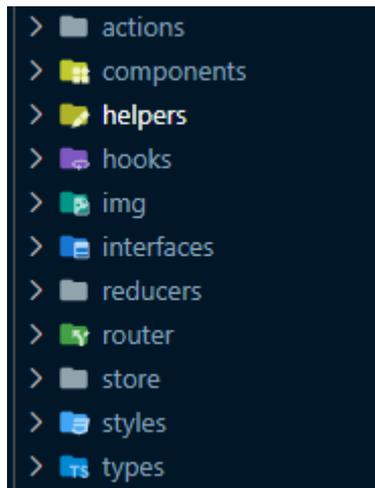


Fig. 28. Estructura de las carpetas frontend

Fuente: Propia

2.3.9. Creación de componentes en el frontend

Los componentes son funciones que aceptan datos de tipo props que son datos de la vista y que retornan elementos de tipo react que detalla lo que debe aparecer en pantalla.

Los componentes permiten separar la interfaz de usuario en partes totalmente independientes una de otra, reutilizables y pensar en cada pieza de forma aislada. (Componentes y Propiedades – React, n.d.)

En este bloque se realiza la creación de los componentes que vamos a usar a lo largo de todo el proyecto, en la Fig. 29 se da un ejemplo de la estructura básica que se usó en todos los componentes.

```

const NavBar = () => {
  const dispatch = useDispatch();
  const handleLogout = () => dispatch(startLogout());

  return (
    <div>
      <div className="navbar-fixed">
        <nav>
          <div className="nav-wrapper fixed black">
            <a href="#" className="brand-logo hide-on-med-and-down">
              I17J
            </a>
            <ul id="nav-mobile" className="right">
              <li>
                <Link className="navbar-item" to="/">
                  Convocatoria
                </Link>
              </li>
              <li>
                <Link className="navbar-item" to="/actas">
                  Actas
                </Link>
              </li>
              <li className="navbar-item" onClick={handleLogout}>
                Logout
              </li>
            </ul>
          </div>
        </nav>
      </div>
    </div>
  );
};

```

Fig. 29. Componente barra navegación

Fuente: Propio

2.3.10. Creación de reducers en el frontend

En Redux, un reducer es una función que se encarga de una acción y guarda un estado anterior de la aplicación y devuelve un nuevo estado. Una acción describe lo que sucedió y es trabajo del reducer devolver el nuevo estado basado en esa acción. (Reducers - Redux En español, n.d.)

En la Fig. 30 se puede ver un reducer que se encarga de cambiar el estado de cuando una persona inicia o cierra sesión, este reducer es un ejemplo básico de todos los reducer usados en el desarrollo del sistema.

```

const AuthReducer = (state = init, action:i_action):i_state => {

  const { login, logout,checking } = types;
  const { type, payload } = action;

  switch( type ) {
    case login :
      state = {
        ...state,
        ...payload
      };
      break;
    case checking:
      state = {
        ...state,
        checking: false
      }
      break;
    case logout:
      state = {
        ...init,
        checking: false
      };
      break;
  }

  return state;
}

```

Fig. 30. Reducer de inicio y cierre de sesión

Fuente: Propia

2.3.11. Creación de rutas en el frontend

En este bloque se crea las rutas con las cuales podremos navegar entre componentes dependiendo de la acción del usuario. En la Fig. 31 tenemos la creación de las rutas en el caso de que una persona se halla logeado de manera correcta entra al sistema o en el caso de haber introducido mal las credenciales el usuario es redireccionado automáticamente a la página principal.

```

const Routes = () => {
  const dispatch = useDispatch();
  const { uid,checking } = useSelector((info:i_redux) => info.auth);

  useEffect(() => {
    dispatch( startChecking() );
  },[dispatch]);

  const isAuth = !!uid;

  if( checking ) return <Loading />

  return <Router>
    <Switch>
      <PublicRouter
        isAuthenticate={ isAuth }
        path='/login'
        exact={ true }
        Component={ AuthScreen }
      />
      <PrivateRouter
        isAuthenticate={ isAuth }
        path='/'
        exact={ false }
        Component={ RootRoutes }
      />
    </Switch>
  </Router>
}

export default Routes;

```

Fig. 31. Rutas login

Fuente: Propia

2.3.12. Creación de interfaces en el backend y frontend

TypeScript hace la verificación de tipos de datos y se centra en la forma en que tienen estos valores, las interfaces cumplen la función de nombrar estos tipos y forzar a que los datos se presenten de la manera en que fueron definidos. (TypeScript: Handbook - Interfaces, n.d.)

En la Fig. 32 se muestra un ejemplo de interfaz básica que se usa en el desarrollo del proyecto, esta asegura que el tipo de datos que se requiere venga de la forma en la que se definió el dato.

```
export interface i_meeting {
  nombre: string
  link: string
  img: string
  fecha: Date
  detalle: string
  title: string
  platform: string
}
```

Fig. 32. Interfaz meeting

Fuente: Propia

2.4. Pruebas

2.4.1. Pruebas de carga con Postman

Postman es un software que nos ayuda a consumir peticiones http mediante las cuales podemos realizar pruebas de carga de API's. al consumir una API se configuro en el backend diversos códigos en el caso de algún error en el tiempo de carga al consumir el API

- Http 500: Erro inesperado por parte del servidor.
- Http 400: El dato solicitado no existe.
- Http 404: No se encuentra el dato solicitado.
- Http 200: Respuesta exitosa el servicio fue ejecutado sin problema.

2.4.1.1. Consumo y validación de los servicios API

En Postman tenemos la posibilidad de agrupar diferentes requests. A esta capacidad de Postman se le conoce como colecciones y nos ayuda a organizar nuestras pruebas en la Fig. 33 imagen podemos ver la colección de pruebas.

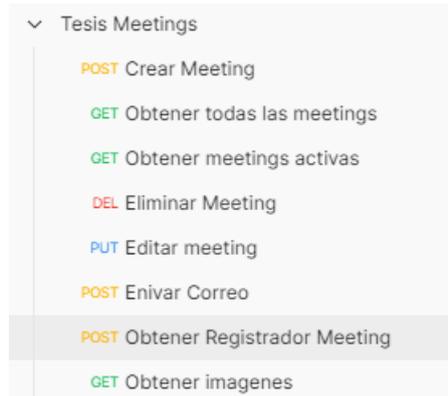


Fig. 33. Colección http

Fuente: Propia

Para consumir y probar un API se realiza mediante una URL que dependiendo del servicio HTTP y su configuración en el backend arroja distintos resultados en la Fig. 34 se puede ver la introducción de la URL y la introducción de datos y en la Fig. 35 se puede ver el resultado del consuma del API y el tiempo que se demora en consumir la URL.

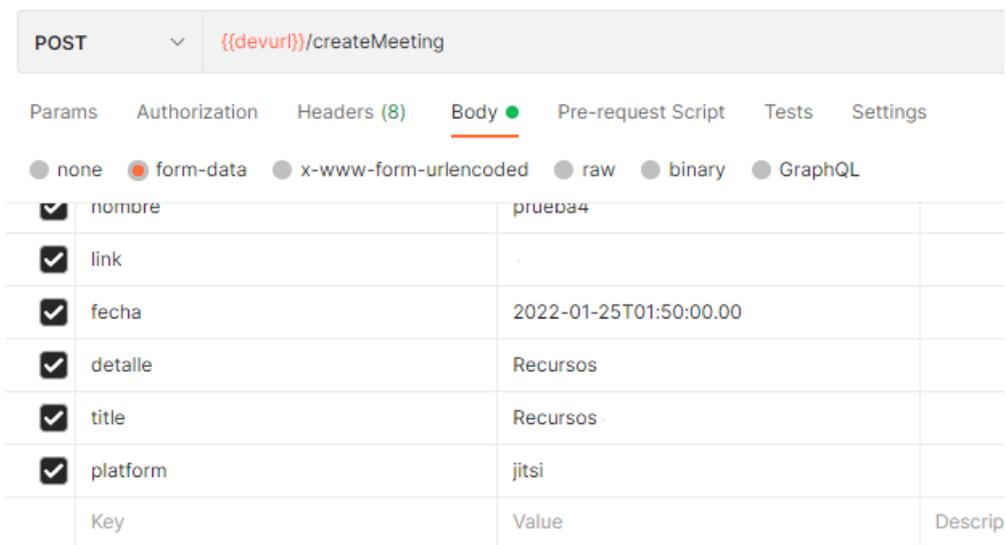


Fig. 34. Inserción de URL para a consumir

Fuente: Propia

```
Body Cookies Headers (21) Test Results 200 OK 237 ms 1.21 KB
Pretty Raw Preview Visualize JSON
1
2 "ok": true,
3 "message": "Meeting creada y guardada con éxito",
4 "data": {
5     "nombre": "prueba4",
6     "link": " ",
7     "img": " ",
8     "fecha": "2022-02-25T06:50:00.000Z",
9     "detalle": "Recursos",
10    "title": "Recursos ",
11    "platform": "jitsi",
12    "_id": "61f82e1703f00dd3fdc8d625",
13    "__v": 0
14 }
15
```

Fig. 35. Resultado de petición http

Fuente: Propia

2.4.2. Pruebas unitarias

Las pruebas unitarias consisten en aislar pequeñas porciones de código y verificar que estas funcionen de manera correcta, de esta manera aseguramos una correcta funcionalidad del software. Para las pruebas unitarias se hizo uso de la librería Jasmine

Jasmine es un marco de pruebas de desarrollo impulsado por el comportamiento para JavaScript. (Jasmine Para JavaScript, n.d.).

- Configuración de archivos a testear

Para poder analizar de manera rápida todas las funciones del código en el archivo de configuración de Jasmine especificamos el directorio de la carpeta de los servicios del backend y frontend. En la Fig. 37 se puede ver que se va a probar todo el código que se encuentre dentro de la carpeta service.

```

{
  "spec_dir": "spec",
  "spec_files": [
    "**/*[sS]pec.?(m)js"
  ],
  "helpers": [
    "helpers/**/*.?(m)js"
  ],
  "services": [
    "services/**/*.?(m)js"
  ],
  "env": {
    "stopSpecOnExpectationFailure": false,
    "random": true
  }
}

```

Fig. 36. Configuración de jasmine pruebas unitarias

Fuente: Propia

- Inicialamos la librería Jasmine

Una vez especificado los archivos a testear, se procede a configurar el código que iniciará las pruebas de cada función. En la Fig. 37 se puede ver una parte del código para las pruebas.

```

let Jasmine = require('jasmine');
let jasmine = new Jasmine();

jasmine.loadConfigFile('spec/support/jasmine.json');
jasmine.configureDefaultReporter({
  showColors: false
});
jasmine.execute();

```

Fig. 37. Código para inicio de pruebas unitarias

Fuente: Propia

- Ejecución del código y resultados

En la Fig. 38 se puede ver que las pruebas unitarias fueron ejecutadas y finalizadas con éxito.

```
$ node test
Iniciando prueba
Test 1...
Test 2...
Test 3...
Test 4...
Los archivos de las carpeta service funcionan de manera correcta
```

Fig. 38. Resultado pruebas unitarias

Fuente: Propia

CAPÍTULO 3

3. Resultados

3.1. Validación de resultados mediante la ISO 25010

La ISO 25010 analiza la calidad del sitio web utilizando métodos estándar para evaluar la calidad de uso prestando atención a la perspectiva del usuario, incluida la eficacia, la eficiencia, el nivel de satisfacción del sitio web en satisfacer las necesidades de los usuarios.(Wulandari & Aristana, 2021)

El software fue validado con la subcaracterística de usabilidad, el análisis de la calidad desde el punto de vista de las pruebas de usabilidad se llevó a cabo utilizando un método de cuestionario que hace referencia al System Usability Scale de Hedelefs Aguilar y Garza Villegas (2016).

3.2. Metodología

En el presente proyecto para la evaluación de usabilidad del módulo de videoconferencias se usó la una versión del cuestionario de 10 preguntas adaptado al español de (Hedelefs Aguilar y Garza Villegas, 2016).

El cuestionario este compuesto de las siguientes preguntas:

1. ¿Cree que me gustaría utilizar frecuentemente este sitio web?
2. ¿Encontró el sitio web sencillo?
3. ¿Piensa que el sitio web es fácil de usar?
4. ¿Piensa que podría utilizar este sitio web sin el apoyo de un técnico?
5. ¿Encontró que varias de las funciones en el sitio web estaba bien integradas?
6. ¿Piensa que había demasiada consistencia en el sitio web?
7. ¿Imagina que la mayoría de las personas podrían aprender a usar este sitio web muy rápido?
8. ¿Encontró el sitio web muy intuitivo?
9. ¿Se sintió muy confiado (seguro) al utilizar el sitio web?

10. ¿Puede utilizar el sitio web sin tener que aprender algo nuevo?

La respuesta se ha realizado en la escala de Likert, en la TABLA 16 se ha puesto como puntaje a 5 totalmente de acuerdo y 1 totalmente en desacuerdo según Sauro (2016).

TABLA 16. Puntuación para escala de Likert

Respuesta	Puntaje
Totalmente en desacuerdo	1
En desacuerdo	2
Ni de acuerdo ni, en desacuerdo	3
De acuerdo	4
Totalmente de acuerdo	5

Fuente: Adaptado de (Sauro & Lewis, 2016)

3.3. Evaluación

La encuesta se realizó a través de la plataforma forms de office 365 a estudiantes de 3 niveles diferentes de la carrera de Ingeniería de Software, con el fin de que los estudiantes puedan interactuar directamente con el sistema de videoconferencias, el software se desplego a producción en digital ocean plataforma que sirve para la creación servidores y desplegar aplicaciones, para dar más seguridad al sistema se hizo la adquisición de un dominio y un certificado SSL.

Todo el proceso se realizó a través de la plataforma teams con un total de 58 estudiantes, los resultados de las encuestas se pueden observar en la TABLA 17 estructurado de manera que en la primera fila se observan las respuestas y en la primera columna se visualiza en orden las preguntas del cuestionario.

TABLA 17.
Resultados de la encuesta por pregunta

Preguntas	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo, ni en desacuerdo	De acuerdo	Totalmente de Cuerto
Pregunta 1	2	1	7	8	40
Pregunta 2	2	1	5	11	39
Pregunta 3	2	0	5	12	39
Pregunta 4	2	2	5	9	40
Pregunta 5	2	1	5	7	43
Pregunta 6	2	0	9	8	39
Pregunta 7	1	1	4	8	44
Pregunta 8	2	1	5	11	39
Pregunta 9	2	4	5	5	42
Pregunta 10	1	2	5	6	44

3.4. Análisis de resultados

A continuación, se detalla el resultado obtenido da cada pregunta de la encuesta SUS aplicada a los estudiantes.

Pregunta 1: Creo que me gustaría utilizar frecuentemente este sitio web.

En la Fig. 39 que corresponde a la pregunta 1 y en su análisis se puede observar que la opción más votada fue “Total mente de acuerdo” con un 69% de los votos, mientras que la

segunda opción más votada fue “De acuerdo” con un 14% de los votos. Este resultado muestra que, los usuarios tendrían una gran propensión a usar el sitio web de manera repetitiva.

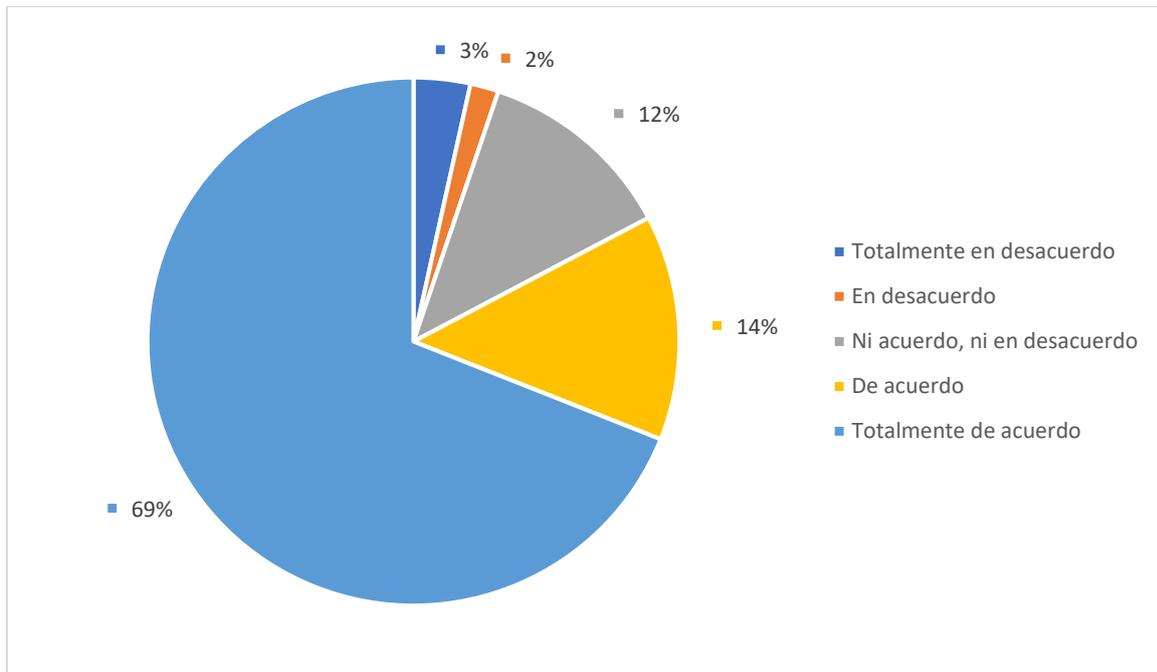


Fig. 39. Resultados de la pregunta 1 de la encuesta SUS

Fuente: Propia

Pregunta 2: Encontré el sitio web sencillo.

En la Fig. 40 que corresponde a la pregunta 2 y en su análisis se puede observar que la opción más votada fue “Total mente de acuerdo” con un 67% de los votos, mientras que la segunda opción más votada fue “De acuerdo” con un 19% de los votos. Este resultado muestra que, los usuarios una interfaz bastante sencilla de entender.

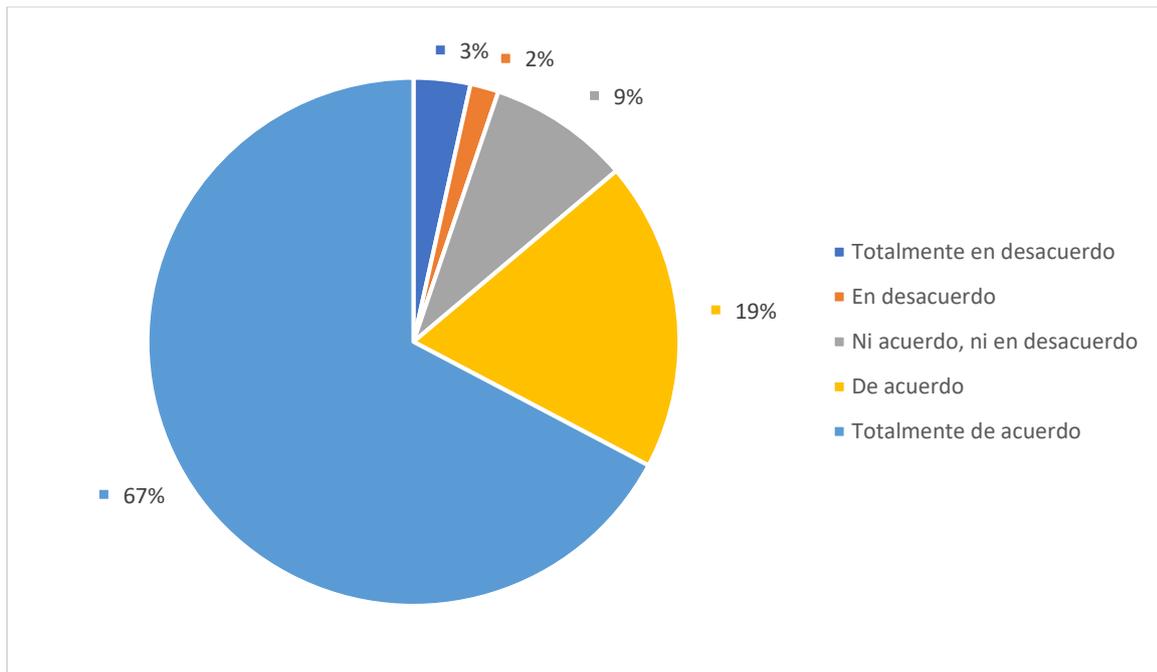


Fig. 40. Resultados de la pregunta 2 de la encuesta SUS

Fuente: Propia

Pregunta 3: Pienso que el sitio web es fácil de usar.

En la Fig. 41 que corresponde a la pregunta 3 y en su análisis se puede observar que la opción más votada fue “Total mente de acuerdo” con un 67% de los votos, mientras que la segunda opción más votada fue “De acuerdo” con un 21% de los votos. Este resultado muestra que, los usuarios encontraron el sitio web muy interactivo y fácil de usar, incluso para personas que no estén acostumbradas a usar plataformas de videoconferencias.

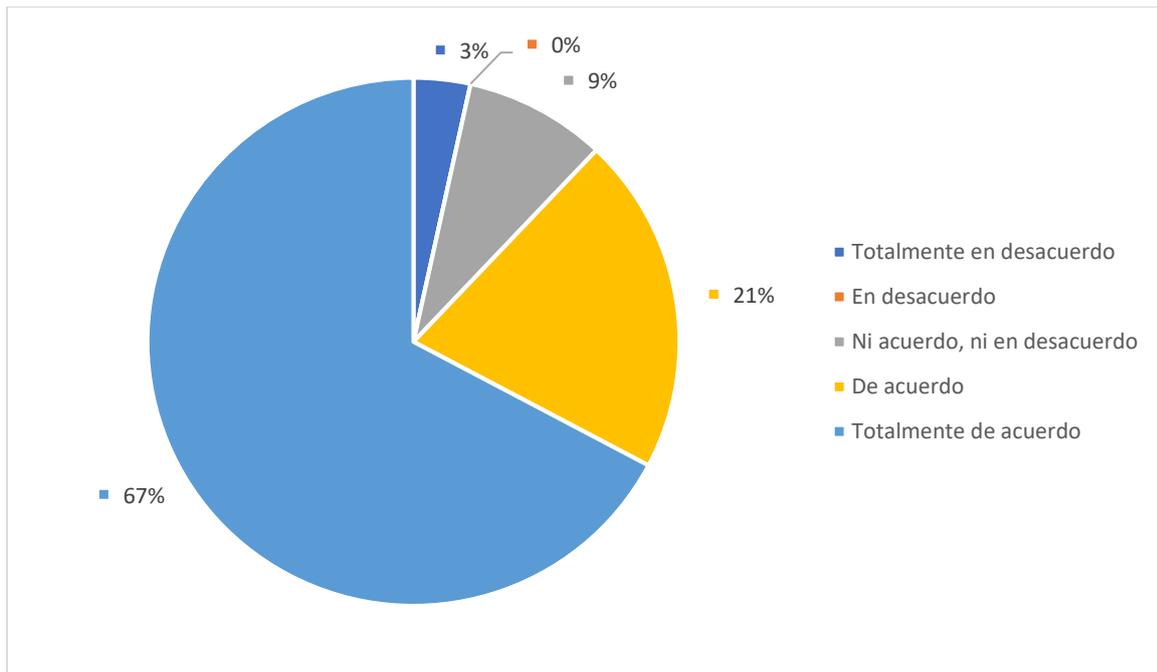


Fig. 41. Resultados de la pregunta 3 de la encuesta SUS

Fuente: Propia

Pregunta 4: Pienso que podré utilizar este sitio web sin el apoyo de personal técnico.

En la Fig. 42 que corresponde a la pregunta 4 y en su análisis se puede observar que la opción más votada fue “Total mente de acuerdo” con un 69% de los votos, mientras que la segunda opción más votada fue “De acuerdo” con un 16% de los votos. Este resultado muestra que, que los usuarios necesitan poca o ninguna ayuda para interactuar con el sitio web.

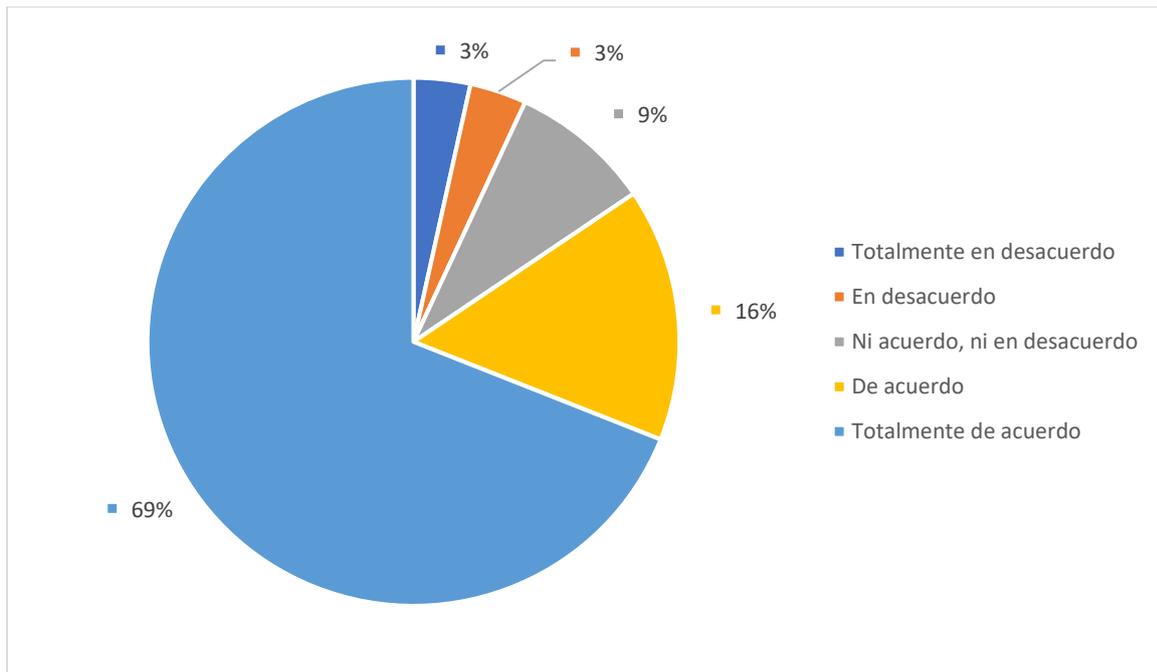


Fig. 42. Resultados de la pregunta 4 de la encuesta SUS

Fuente: Propia

Pregunta 5: Encontré que varias de las funciones en el sitio en el sitio web estaban bien integradas.

En la Fig. 43 que corresponde a la pregunta 5 y en su análisis se puede observar que la opción más votada fue “Total mente de acuerdo” con un 74% de los votos, mientras que la segunda opción más votada fue “De acuerdo” con un 12% de los votos. Este resultado muestra que, todos los componentes usados en el software son funcióneles y que no existe problema en la interacción entre las funciones del software.

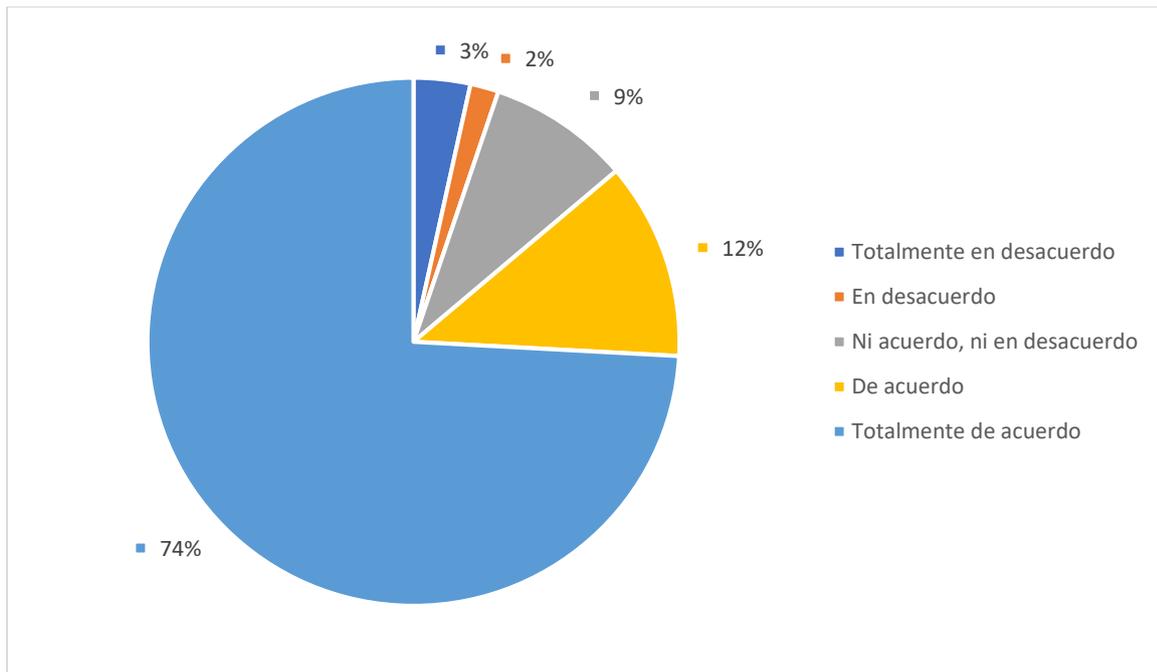


Fig. 43. Resultados de la pregunta 5 de la encuesta SUS

Fuente: Propia

Pregunta 6: Pensé que había demasiada consistencia en el sitio web.

En la Fig. 44 que corresponde a la pregunta 6 y en su análisis se puede observar que la opción más votada fue “Total mente de acuerdo” con un 67% de los votos, mientras que la segunda opción más votada fue “De acuerdo” con un 14% de los votos. Este resultado muestra que, el sitio web tienen es estable y coherente al momento de mostrar información requerida.

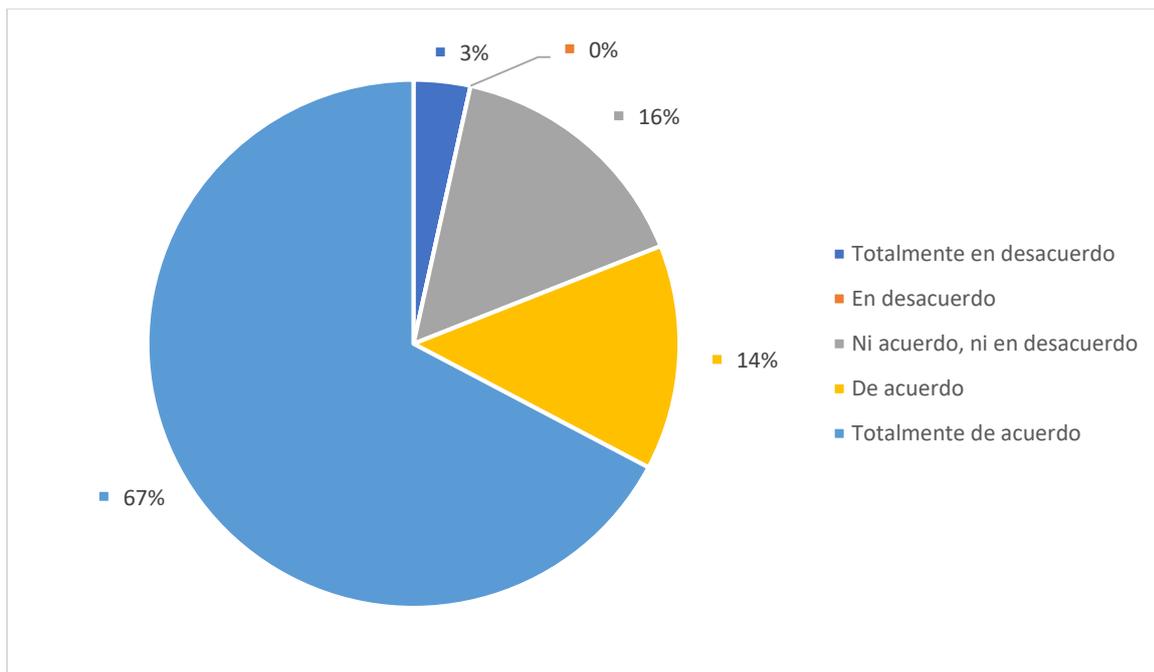


Fig. 44. Resultados de la pregunta 6 de la encuesta SUS

Fuente: Propia

Pregunta 7: Me imagino que la mayoría de las personas podrían aprender a usar este sitio web muy rápido.

En la Fig. 45 que corresponde a la pregunta 7 y en su análisis se puede observar que la opción más votada fue “Total mente de acuerdo” con un 76% de los votos, mientras que la segunda opción más votada fue “De acuerdo” con un 14% de los votos. Este resultado muestra que, personan no expertas podrían usar el sitio web con facilidad.

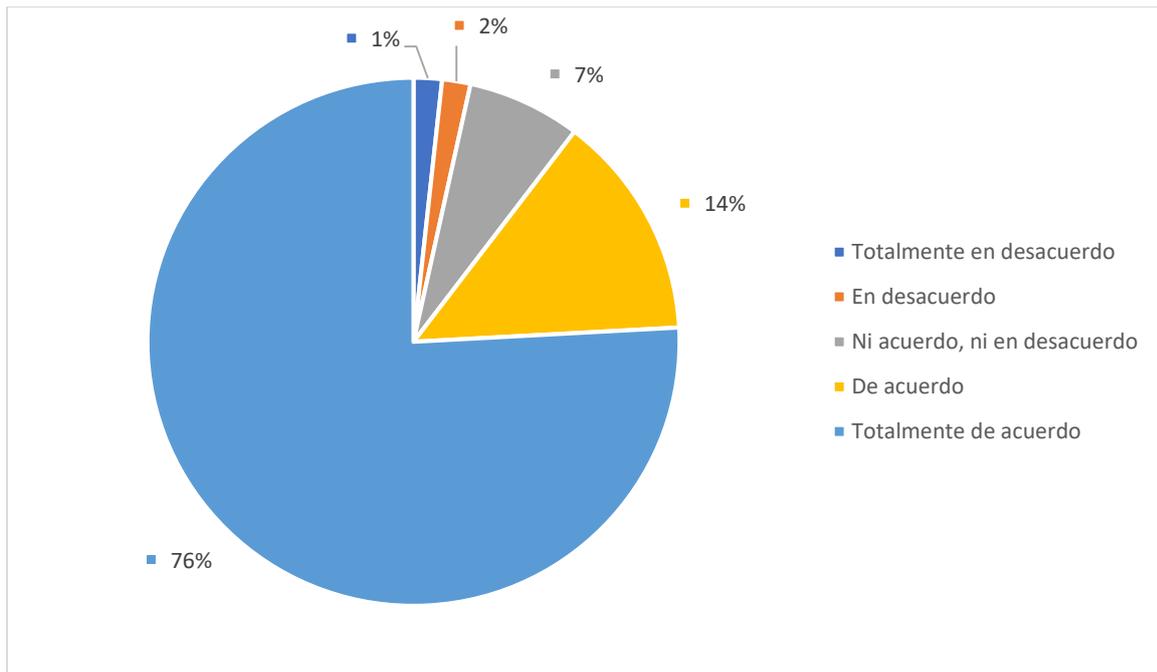


Fig. 45. Resultados de la pregunta 7 de la encuesta SUS

Fuente: Propia

Pregunta 8: Encontré el sitio web muy intuitivo.

En la Fig. 46 que corresponde a la pregunta 7 y en su análisis se puede observar que la opción más votada fue “Total mente de acuerdo” con un 67% de los votos, mientras que la segunda opción más votada fue “De acuerdo” con un 19% de los votos. Este resultado muestra que, el usuario muestra una clara apreciación de lo que debe hacer al momento de ejecutar alguna acción dentro del sistema web.

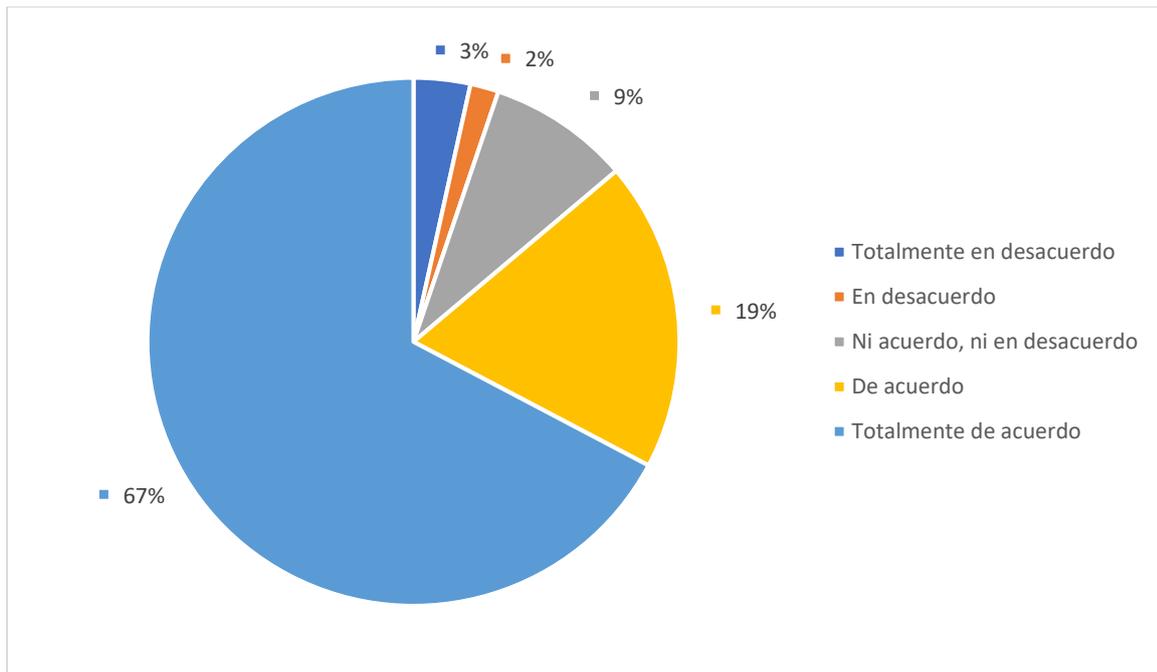


Fig. 46. Resultados de la pregunta 8 de la encuesta SUS

Fuente: Propia

Pregunta 9: Me sentí muy confiado (seguro) al utilizar el sitio web.

En la Fig. 47 que corresponde a la pregunta 7 y en su análisis se puede observar que la opción más votada fue "Total mente de acuerdo" con un 72% de los votos, mientras que la segunda opción más votada fue "De acuerdo" con un 9% de los votos. Este resultado muestra que, el usuario siente confianza es el uso del sistema su experticia no ha sido afectada por algún error.

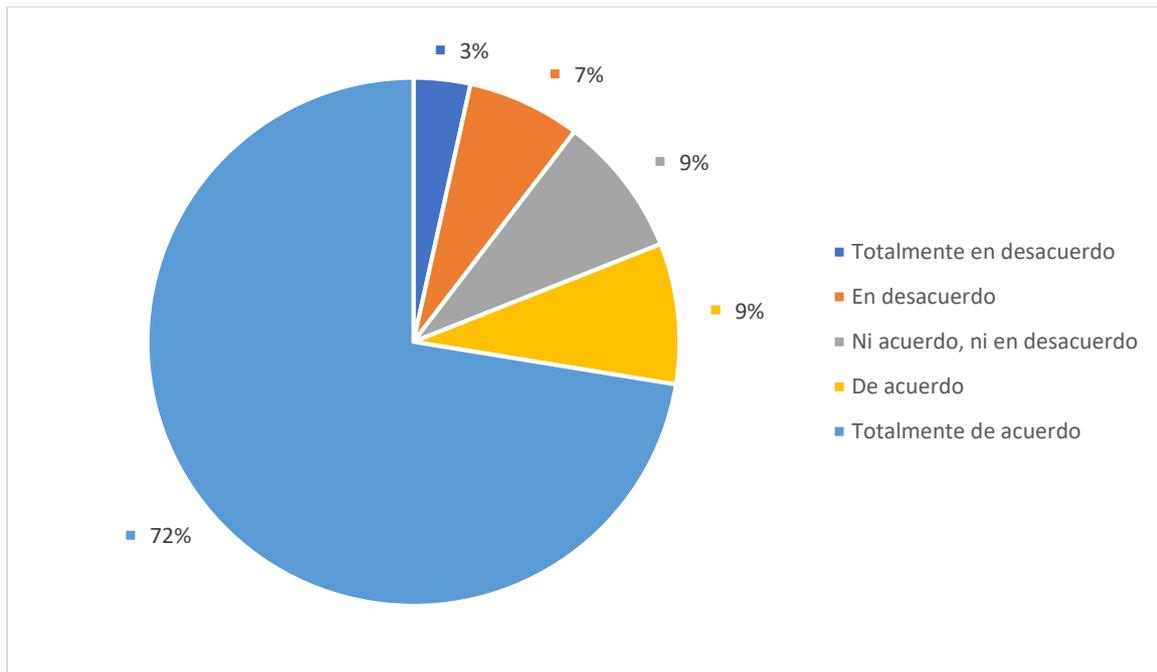


Fig. 47. Resultados de la pregunta 9 de la encuesta SUS

Fuente: Propia

Pregunta 10: Pude utilizar el sitio web sin tener que aprender nada nuevo.

En la Fig. 48 que corresponde a la pregunta 7 y en su análisis se puede observar que la opción más votada fue "Total mente de acuerdo" con un 72% de los votos, mientras que la segunda opción más votada fue "De acuerdo" con un 9% de los votos. Este resultado muestra que, el uso del sitio de videoconferencias es fácil de entender y no necesita de mucho conocimiento acerca de aplicaciones de videoconferencia para aprender a usar el software.

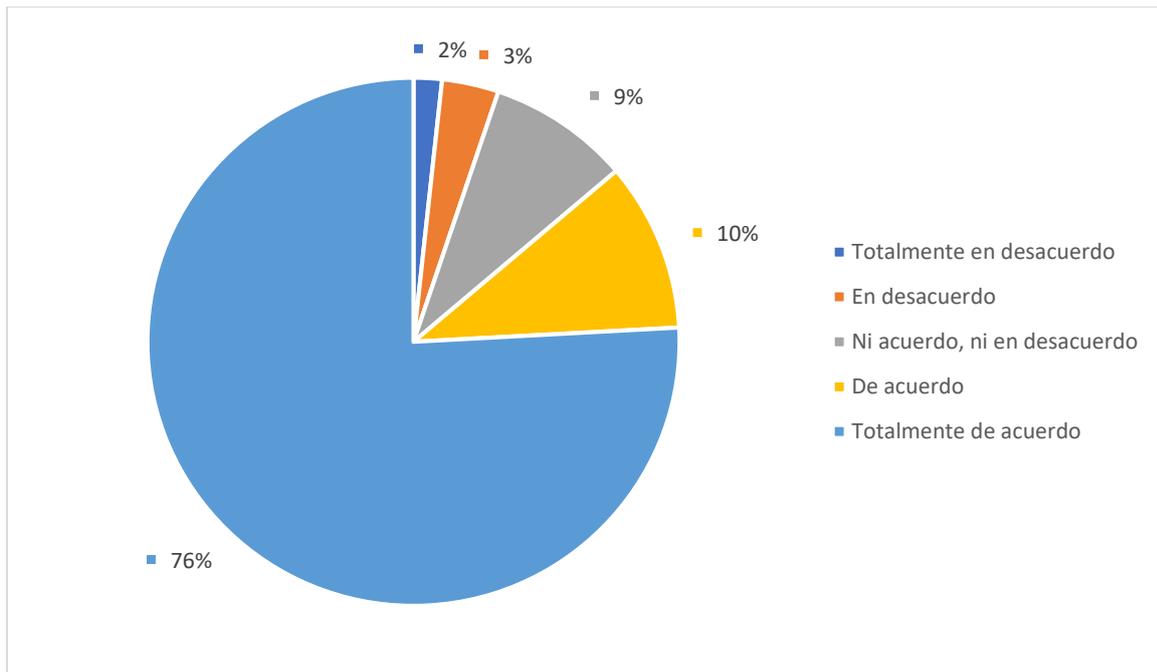


Fig. 48. Resultados de la pregunta 10 de la encuesta SUS

Fuente: Propia

3.5. Interpretación de resultados

En este bloque se analiza la encuesta (SUS) para obtener el resultado de la usabilidad del software. Para esto hacemos uso de la escala de Likert visto en la TABLA 16 y el método para valorar las opciones se dividen en preguntas pares donde 1 es “Totalmente en desacuerdo” y 5 es “Totalmente de acuerdo” y para preguntas impares la puntuación se invierte donde 1 es “Totalmente de acuerdo” y 5 es “Totalmente en desacuerdo”, en TABLA 18 la se da a conocer más a detalle cada uno de los casos.

TABLA 18.
Valoración de las respuestas de escala SUS

Opciones	Preguntas pares	Preguntas impares
Totalmente en desacuerdo	5	1
En desacuerdo	4	2
Ni de acuerdo, ni en desacuerdo	3	3
De acuerdo	2	4
Totalmente de acuerdo	1	5

Tomamos cada una de las encuestas individualmente y se procede a dar el puntaje Teniendo en cuenta la valoración de las respuestas de la escala SUS vista en la TABLA 18 en la () se indexo el resultado de cada una de las encuestas clasificado por pregunta y numero de encuestado.

TABLA 19.
Puntuación individual de cada encuesta

Encuesta	Preguntas									
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	5	2	5	1	4	1	5	2	1	5
2	5	1	5	1	5	3	5	1	1	5
3	5	1	5	1	5	1	5	1	1	5
.
.
.
56	3	3	5	2	2	2	5	2	4	3
57	5	1	5	1	5	1	5	1	1	5
58	5	1	5	1	5	1	5	1	1	5

Una vez colocados los valores de cada respuesta individualmente de cada encuestado, se hace una sumatoria de las preguntas pares(Y) y las preguntas impares(X) de cada encuesta, en la TABLA 20 se puede observar la sumatoria de las preguntas pares he impares de cada uno de los encuestados aplicando la siguiente formulas $X=P1+P3+P5+P7+P9$ y $Y=P2+P4+P6+P8+P10$

TABLA 20.
Sumatoria Preguntas pares e impares

Encuesta	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	X	Y
1	5	2	5	1	4	1	5	2	5	1	24	7
2	5	1	5	1	5	3	5	1	5	5	25	11
3	5	1	5	1	5	1	5	1	5	5	25	9
.
.
.
56	3	3	5	2	2	2	5	2	2	3	17	12
57	5	1	5	1	5	1	5	1	5	1	25	5
58	5	1	5	1	5	1	5	1	5	1	25	5

Una vez hecho la sumatoria de las preguntas pares(Y) e impares (X), se aplica las siguientes formulas $X0=X-5$ y $Y0=25-Y$, estas 2 datos son muy importantes para poder obtener el grado SUS aplicando la siguiente formula $SUS=(X0+Y0)*2.5$. En la TABLA 21 se puede ver el grado SUS de cada encuesta y el promedio SUS total de todas la encuesta.

TABLA 21.
Promedio de la escala SUS

Encuesta	X	Y	X0	Y0	SUS
1	24	7	19	18	92.5
2	25	11	20	18	95
3	25	9	20	20	100
.
.
.
56	17	14	12	11	57.5
57	25	5	20	20	100
58	25	5	20	20	100
Promedio					86.81

El promedio de las 58 encuestas realizadas a los estudiantes de la carrera de ingeniería en software es de 86.67 que arrojan un grado B que significa que la usabilidad del software es buena y acptable en la Fig. 49 podemos observar la escala de calificación SUS.

System Usability Score



Fig. 49. Grado de calificación SUS

Fuente: Adaptado de (The System Usability Scale & How It's Used in UX | Adobe XD Ideas, 2021.)

CONCLUSIONES

Se logró consolidar los procedimientos que enmarcan los procesos académicos ejecutados al interior de la empresa, lo cual representa una disminución en los tiempos de las actividades, así como la optimización de los servicios ofertados, lo cual genera un aumento en la calidad del proceso de aprendizaje.

La revisión literaria permitió tener información sólida y efectiva que fue útil para realizar el marco teórico, en lo referente a tecnologías y herramientas que se necesitan para el desarrollo de la aplicación web.

La Implementación del software sobre plataformas web permite interactuar de manera fácil y sencilla con otros módulos del sistema EVAI, además posibilita la comunicación directa con el cliente y el administrador del sistema.

Ante la necesidad de mejorar y fortalecer los procesos académicos, se hizo uso de la ISO-25010 que permitió ver el grado usabilidad del software mediante la creación de encuestas, que por medio del método de system usability score, que facilitó el proceso de evidenciar los resultados obtenidos en el capítulo de validación de resultados, también ayudo a evidenciar que procesos mejoraron y fueron fortalecidos y que otros procesos deben mejorar.

RECOMENDACIONES

La encuesta de usabilidad SUS debe ser usadas para encuestas que contengan 10 preguntas y 5 respuestas, porque su análisis matemático y estadístico está elaborado para este rango de respuestas. Si se desea ampliar las preguntas y repuesta se debería buscar otra metodología.

Digital Ocean es un excelente servidor para alojamiento de cualquier tipo de aplicación, en este tipo de servidores ese necesario configurar todo desde cero debido a que es una máquina virtual en un sistema operativo Linux, se sugiere realizar investigaciones de servicios en la nube donde únicamente se deba desplegar la aplicación y que estos servicios se encarguen de la seguridad del servidor.

Se sugiere el uso de la metodología XP solo en proyectos a corto plazo ya que tiene tanto ventajas como desventajas, entre los beneficios se destacan la satisfacción del cliente, programación organizada e implementación una forma de trabajo donde se adapta fácilmente a la circunstancia, entre las desventajas se encuentran que XP es una metodología que trabaja de forma rápida y de cambios constantes que da una carga más pesada trabajo a todos los roles.

REFERENCIAS

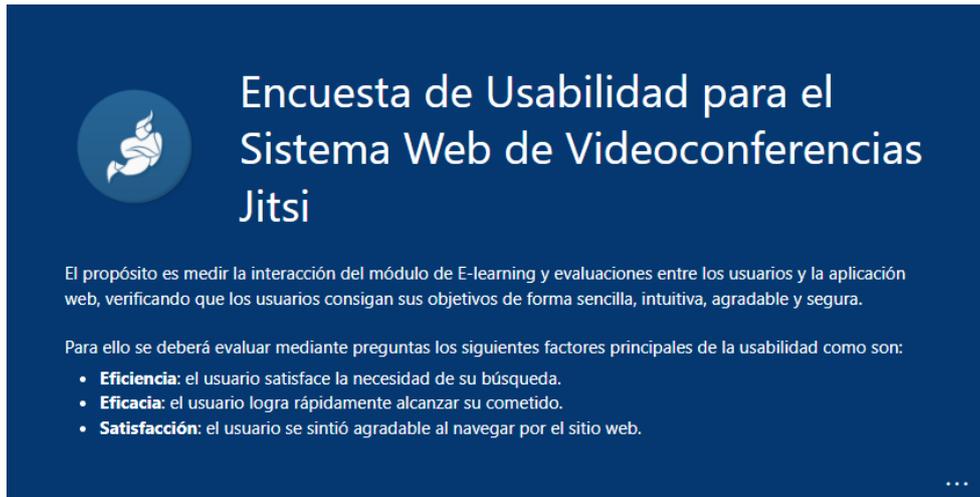
- Aguirre, F., & Kevin, I. (2021). Diseño de un sistema guía de rutas basado en la metodología mobile-d para la movilidad de personas con discapacidad visual en el centro de la ciudad de Ibarra. <http://repositorio.utn.edu.ec/handle/123456789/11639>
- Architecture · Jitsi Meet Handbook. (n.d.). Retrieved November 10, 2021, from <https://jitsi.github.io/handbook/docs/architecture#architecture-diagram>
- Cachimuel, B. (2020). Estudio comparativo sobre bases de datos NOSQL, que permitan analizar la velocidad de respuesta de los datos en un prototipo de red social universitaria. <http://repositorio.utn.edu.ec/handle/123456789/10365>
- Castro, J. (2014). IMPLEMENTACIÓN DEL SISTEMA AUTOMATIZADO DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE. <http://repositorio.utn.edu.ec/handle/123456789/10836>
- Castro Querembas, J. A. (2020). Diseño de un ecosistema de software, para la interoperabilidad entre sistemas de E-Commerce y Courier mediante Apis Restful Eficientes y Seguras. <http://repositorio.utn.edu.ec/handle/123456789/10836>
- Coloma, A. C., Barco, G. C., & Lescano, L. A. R. (2017). Técnicas para el uso de la metodología eXtreme Programming en el desarrollo de Software. Hacedor - AIAPÆC, 1(1). <http://revistas.uss.edu.pe/index.php/HACEDOR/article/view/499>
- Componentes y propiedades – React. (n.d.). Retrieved January 26, 2022, from <https://es.reactjs.org/docs/components-and-props.html>
- Esquivel Vega, G., Treviño Villalobos, M., Viquez Acuña, L., & Quirós Oviedo, R. (2019). Una comparación de rendimiento entre bases de datos NoSQL: MongoDB y ArangoDB. Tecnología En Marcha, ISSN 0379-3962, ISSN-e 2215-3241, Vol. 32, N°. Extra-9, 2019 (Ejemplar Dedicado a: VIII Encuentro de Investigación y Extensión), Págs. 5-15, 32(9), 5–15. <https://doi.org/10.18845/tm.v32i6.4223>
- Express. (n.d.). Retrieved January 13, 2022, from <https://expressjs.com/es/guide/using-middleware.html>
- Giordani, J., Vargas, C., Pierre, J., Limaco, P., Manuel, I., & Mondragón Vilela, R. (2021). Estudio comparativo entre las bases de datos relacional y no relacionales: una revisión de la literatura científica. Universidad Privada Del Norte. <https://repositorio.upn.edu.pe/handle/11537/27388>
- Introduction · Jitsi Meet Handbook. (n.d.). Retrieved November 3, 2021, from <https://jitsi.github.io/handbook/docs/intro>
- Introduction to Node.js. (n.d.). Retrieved December 8, 2021, from <https://nodejs.dev/learn/introduction-to-nodejs>
- Jasmine para-JavaScript. (n.d.). Retrieved January 26, 2022, from <https://jasmine.github.io/setup/nodejs.html>
- MongoDB: the application data platform | MongoDB. (n.d.). Retrieved December 2, 2021, from <https://www.mongodb.com/>

- Palos-Sánchez, P. R., Arenas-Márquez, F. J., & Aguayo-Camacho, M. (2017). La adopción de la tecnología cloud computing (SaaS): Efectos de la complejidad tecnológica vs formación y soporte. *RISTI - Revista Iberica de Sistemas e Tecnologías de Informacao*, 22, 89–105. <https://doi.org/10.17013/RISTI.22.89-105>
- Qué es Node.js y por qué deberías usarlo. (n.d.). Retrieved December 8, 2021, from <https://kinsta.com/es/base-de-conocimiento/que-es-node-js/>
- React – Una biblioteca de JavaScript para construir interfaces de usuario. (n.d.-a). Retrieved December 8, 2021, from <https://es.reactjs.org/>
- Reducers · Redux en Español. (n.d.). Retrieved January 26, 2022, from <https://es.redux.js.org/docs/basico/reducers.html>
- Sauro, J., & Lewis, J. R. (2016). Standardized usability questionnaires. Quantifying the User Experience, 185–248. <https://doi.org/10.1016/B978-0-12-802308-2.00008-4>
- TypeScript: Handbook - Interfaces. (n.d.). Retrieved January 23, 2022, from <https://www.typescriptlang.org/docs/handbook/interfaces.html>
- Wulandari, D. A. P., & Aristana, M. D. W. (2021). Analysis Evaluation Management Information System Audit Internal Quality. *Journal of Electrical, Electronics and Informatics*, 5(1), 5–9. <https://doi.org/10.24843/JEEI.2021.V05.I01.P02>
- Yagual, R. C., Magdalena, L., & Versoza, P. (2018a). Análisis de las ventajas y desventajas del Sistema de Gestión de la Seguridad de la Información y su influencia en la competitividad de las empresas que utilizan Cloud Computing y Big Data en el Ecuador. *INNOVA Research Journal*, 3(4), 181–195. <https://doi.org/10.33890/INNOVA.V3.N4.2018.562>

ANEXOS

Anexo A: Encuesta de usabilidad SUS

A continuación, se muestra la encuesta de escala de usabilidad del sistema de videoconferencias.



Encuesta de Usabilidad para el Sistema Web de Videoconferencias Jitsi

El propósito es medir la interacción del módulo de E-learning y evaluaciones entre los usuarios y la aplicación web, verificando que los usuarios consigan sus objetivos de forma sencilla, intuitiva, agradable y segura.

Para ello se deberá evaluar mediante preguntas los siguientes factores principales de la usabilidad como son:

- **Eficiencia:** el usuario satisface la necesidad de su búsqueda.
- **Eficacia:** el usuario logra rápidamente alcanzar su cometido.
- **Satisfacción:** el usuario se sintió agradable al navegar por el sitio web.

	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
Cree que me gustaría utilizar frecuentemente este sitio web.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Encontró el sitio web sencillo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Piensa que el sitio web es fácil de usar.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Piensa que podrá utilizar este sitio web sin el apoyo de personal técnico.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Encontró que varias de las funciones en el sitio web estaban bien integradas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Piensa que había demasiada consistencia en el sitio web.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Imagina que la mayoría de las personas podrían aprender a usar este sitio web muy rápido	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Encontró el sitio web muy intuitivo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Se sintió muy confiado (seguro) al utilizar el sitio web.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Puede utilizar el sitio web sin tener que aprender algo nuevo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>