

**Universidad de las Ciencias Informáticas
Facultad 6**



Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

**“Fabulari, aplicación web de videoconferencia para la Plataforma
Web Interactiva”**

Autores:

Nurisleidis Rojas Rodríguez
Vidal Orlando Acosta García

Tutores:

Ing. Yoandrys S. Pacheco Jeréz
Ing. Yosvany Marco Pérez

La Habana, junio de 2015

“Año 57 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos ser los únicos autores del trabajo **Fabulari, aplicación web de videoconferencia para la Plataforma Web Interactiva** y se autoriza a la Universidad de las Ciencias Informáticas hacer el uso que estimen pertinente con el mismo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Firma del autor

Nurisleydis Rojas Rodríguez

Firma del autor

Vidal Orlando Acosta García

Firma del Tutor

Ing. Yoandrys S. Pacheco Jeréz.

Firma del Tutor

Ing. Yosvany Marco Pérez

Agradecimientos

*Quiero agradecer principalmente a mis padres **Jorge** y **Nuria**, que son lo más grande que tengo en mi vida, los que me convirtieron en la persona que soy hoy.*

A mi papá por apoyarme en mis decisiones y estar junto a mí siempre que lo necesito, por esforzarse tanto día a día para que yo pueda realizar mis sueños.

A mi madre querida, por ser madre, hermana, amiga. No existen palabras para expresar lo agradecida que estoy por todo lo que has hecho por mí, desde darme la vida hasta ser la razón por la que estoy hoy aquí.

*A mi hermanita **Rosi**, gracias por ser parte de mi vida, por dejar de ser la hermanita pequeña y convertirte en mi hermana mayor cuando necesito de ti.*

*A mi abuela **Ana** y mis abuelitos **Tito** y **Minda**, a los tres les agradezco todo lo que han hecho por mí, sus consejos y el amor que me han dado.*

*A mis hermanos **Anisleidis**, **Yosvany** y a esas primas que puedo llamar hermanas, **Yailín**, **Annalie** y **Katia**, gracias pues de una forma u otra han estado siempre apoyándome y ayudándome a seguir adelante.*

*A todos mis tíos y tías, especialmente a **Noel**, **Marilyn** y **Lilian** quienes me han consentido y educado como una hija más.*

*A mis hermanitas de la UCI, **Yeni** y **Gretel**, las que han estado siempre conmigo en los buenos y malos momentos, apoyándome y aconsejándome, convirtiéndose así en parte de mi familia y con ellas mi cuñado **Ángel**, quien siempre se las arregla para darle el toque alegre a los momentos difíciles.*

*A mis verdaderos amigos, aquellos que son y siempre serán amigos. **Adnan**, gracias por tus buenos consejos, esos que me han ayudado a ser una mejor persona, que me han guiado en la carrera, sobre todo en la recta final. A la peque (**Arianna**), por ser mi amiga incondicional y estar pendiente a mí a pesar de la distancia. Al chino (**Jorgito**), por esas llamadas a las 11 de la noche para verificar como iba la tesis.*

*A mi tutor **Pacheco**, por estar ahí siempre que lo necesitábamos, por todos los consejos y recomendaciones dadas.*

*A mi amigo y tutor **Yosvany**, gracias por estar ahí siempre que lo necesitábamos, por las incontables horas que nos dedicaste, por tener siempre el chiste oportuno en mis mayores momentos de estrés, por ser más que tutor, un amigo.*

*Por último y no menos importante a mi **compañero de tesis**, gracias por tener esa paciencia enorme para soportarme, por brindarme su apoyo y esfuerzo y por estar ahí cuando más necesitaba de su ayuda.*

Vidal Orlando

*A mis padres **Silvia** y **Pedro**, por tanto amor, inspiración y sacrificio todos los días.*

*A mis abuelos **Nena** y **Vidal**, mis primeros maestros y guardianes de mi vida.*

*A mi novia **Lisay**, por su comprensión y apoyo, por tanto cariño y entendimiento, por brindarme los más hermosos momentos de mi vida.*

A la familia escogida, mis amigos, por tantos momentos inolvidables, historias vividas y tantas enseñanzas compartidas.

A toda mi familia, que es mi único, verdadero y preciado tesoro.

A mis compañeros de clase, pasillo, preuniversitario, del barrio y los profesores que durante 17 años han construido y aportado tanta experiencia a mi vida, a todos ellos les agradezco por enseñarme a ser un poco mejor cada día.

*A **Yaicel** y **Gustavo** por haberme dedicado su tiempo, en darme apoyo y rectificarme.*

*A mis tutores **Yoandry** y en especial a **Yosvany** por haber dedicado y aguantado tanto tiempo.*

*A mi compañera de tesis **Nuri** por rectificarme cuando me equivoco, sacarme tanto de quicio.*

A todos muchas gracias.

Dedicatoria

Nurisleidis

A mis padres, que les debo todo cuanto he alcanzado.

A mi hermanita, quien llena de alegría cada momento de mi vida.

A toda mi familia, por estar siempre que los necesito.

Vidal Orlando

A mis padres Silvia y Pedro por ser las personas que más quiero en mi vida, en especial a mi mamá por cuidar de mí tan tiernamente y siempre estar pendiente de mi todo el tiempo.

A mi familia por ser un ejemplo de personas que saben ayudarse mutuamente.

Resumen

Las videoconferencias permiten mantener reuniones colectivas entre varias personas que se encuentran en lugares geográficamente distantes, actualmente su uso ha alcanzado un mayor impacto en los diferentes sectores de la sociedad. La investigación realizada permitió desarrollar una aplicación web de videoconferencia para la Plataforma Web Interactiva del Sistema de Transmisión de Canales Virtuales que garantiza la comunicación entre usuarios con el uso del protocolo WebSocket. Para ello se utilizaron tecnologías web como HTML5, CSS3, JavaScript (del lado del cliente), NodeJs (JavaScript del lado del servidor) y la interfaz de programación de aplicaciones WebRTC con el fin de intercomunicar a los usuarios de forma directa sin necesidad de dependencia de terceros. Se obtuvo una aplicación que funciona en una gran diversidad de dispositivos inteligentes como televisores, teléfonos inteligentes, tabletas electrónicas y computadoras personales. Con su uso, la plataforma ofrece alternativas viables de comunicación para modelos de educación a distancia, posibilita realizar intercambios de ideas, experiencias, conocimientos y reuniones entre personas ubicadas en diferentes latitudes definiendo estrategias para solucionar problemas en conjunto.

Palabras clave: NodeJs, Sistema de Transmisión de Canales Virtuales, videoconferencia, WebRTC, WebSocket.

Abstract

Videoconferencing allows maintaining collective meetings between several people who are geographically in distant locations, recently the use of them has reached a major impact on several branches of society. With this research we were able to develop a web conferencing application for the Interactive Web Platform for the Virtual Channels Transmission System that ensured communication between users using the WebSocket protocol. For this web technologies such as HTML5, CSS3, JavaScript (client side), NodeJs (server-side JavaScript) and programming interface application (API, for its acronym in English) WebRTC were used to intercommunicate users directly without third parties. It is obtained an application that works on a wide range of smart devices such as televisions, telephones, electronic tablets and personal computers. With its use, the platform will offer viable alternatives of communication for distance education models, it will enable the exchange of ideas, experiences, knowledge and meetings between people at different latitudes defining strategies to solve problems together.

Keywords: *NodeJs, Virtual Channels Transmission System, videoconferencing, WebRTC*

Introducción	1
Capítulo 1. Fundamentos teóricos sobre la transmisión de videoconferencias en la web	6
1.1 Elementos asociados al dominio del problema.	6
1.1.1 <i>Aplicación web</i>	6
1.1.2 <i>Streaming</i>	6
1.1.3 <i>Videoconferencia</i>	6
1.2 Elementos fundamentales de las videoconferencias	7
1.2.1 <i>Las Redes de comunicaciones</i>	7
1.2.2 <i>La sala virtual de videoconferencia</i>	8
1.2.3 <i>El Codec</i>	9
1.3 Estudio de soluciones existentes	9
1.4 Resultados obtenidos en el trabajo precedente.	11
1.5 Metodología de desarrollo. Proceso Unificado de Desarrollo.....	12
1.6 Herramienta CASE de modelado Visual Paradigm for UML 8.0	13
1.7 Lenguaje de Modelado Unificado (UML)	14
1.8 Entorno de Desarrollo Integrado PhpStorm	14
1.9 Marco de trabajo MEAN	15
1.10 Biblioteca JQuery	16
1.11 API WebRTC	16
1.12 Lenguajes de programación	17
1.13 Protocolo WebSocket	18
1.14 Conclusiones parciales.....	18
Capítulo 2. Análisis y diseño de <i>Fabulari</i>	19
2.1 Modelo de dominio.	19

2.1.1	<i>Descripción general del Modelo de Dominio</i>	19
2.1.2	<i>Descripción de las clases del Modelo de Dominio</i>	19
2.1.3	<i>Diagrama de clases del Modelo de Dominio</i>	20
2.2	Especificación de los requisitos de la aplicación web de videoconferencia, <i>Fabulari</i>	20
2.2.1	<i>Requisitos funcionales</i>	21
2.2.2	<i>Requisitos no funcionales</i>	22
2.3	Descripción de <i>Fabulari</i>	23
2.3.1	<i>Definición de los actores</i>	24
2.3.2	<i>Diagrama de casos de uso del sistema</i>	24
2.3.3	<i>Descripción de casos de uso</i>	25
2.4	Patrón arquitectónico.....	35
2.5	Patrones de diseño de software utilizados para <i>Fabulari</i>	36
2.5.1	<i>Patrones para la asignación de responsabilidades (GRASP)</i>	37
2.5.2	<i>Patrones GoF</i>	38
2.5.2.1	<i>Patrones de Comportamiento</i>	38
2.5.2.2	<i>Patrones Creacionales</i>	39
2.6	Diagramas de secuencia.....	40
2.7	Diagrama de clases del diseño.....	42
2.8	Conclusiones parciales.....	43
Capítulo 3.	Implementación y Pruebas de <i>Fabulari</i>	44
3.1	Diagrama de Componente.....	44
3.2	Estándares de codificación.....	45
3.3	Tratamiento de errores.....	47
3.4	Modelo de despliegue.....	47

3.5	Modelo de Pruebas	48
3.5.1	<i>Tipos de pruebas</i>	49
3.6	Aplicación y resultado de las pruebas	50
3.7	Conclusiones parciales.....	57
	Conclusiones Generales.....	58
	Recomendaciones	59
	Referencias Bibliográficas	60

Índice de Figuras

Figura 1. Representación del Modelo de Dominio.....	21
Figura 2. Diagrama de Caso de Uso.....	26
Figura 3. Vista de la Arquitectura.....	36
Figura 4. Representación del patrón Command.....	39
Figura 5. Representación del patrón de diseño Singleton.....	39
Figura 6. Diagrama de secuencia. Crear Sala de videoconferencia.....	40
Figura 7. Diagrama de secuencia. Listar Sala de videoconferencia.....	40
Figura 8. Diagrama de secuencia. Editar Sala de videoconferencia.....	41
Figura 9. Diagrama de secuencia. Buscar Sala de videoconferencia.....	41
Figura 10. Diagrama de secuencia. Eliminar Sala de videoconferencia.....	42
Figura 11. Diagrama de clases del diseño.....	42
Figura 12. Diagrama de Componente.....	46
Figura 13. Representación de estándares de código utilizados.....	47
Figura 14. Representación de estándares de código utilizados.....	48
Figura 15. Diagrama de Despliegue.....	49
Figura 16. Gráfico de resultado de las No Conformidades.....	53
Figura 17. Representación de la configuración de las Pruebas de resistencia.....	55
Figura 18. Representación de resultados de las Pruebas de resistencia con 5 muestras.....	56
Figura 19. Representación de resultados de las Pruebas de resistencia con 50 muestras.....	56
Figura 20. Representación de resultados de las Pruebas de resistencia con 1000 muestras.....	57

Figura 21. Representación de resultados de las Pruebas de resistencia. Resultados generales con 1000 muestras.....57

Índice de Tablas

Tabla 1. Tabla sinóptica de estudio de homólogos.....11

Tabla 2. Descripción de los actores del sistema.....24

Tabla 3. Descripción de caso de uso: Administrar salas de videoconferencia.....25

Tabla 4. Descripción de caso de uso: Visualizar videoconferencia.....31

Tabla 5.Descripción de variables del Caso de Prueba del CU Administrar sala de videoconferencia.....49

Tabla 6. DCP para el CU: Administrar Sala de videoconferencia. Sección: Crear sala de videoconferencia.....50

Introducción

A través de los años, el hombre fue desarrollando las comunicaciones, apegado a una evolución tecnológica y surgida de una necesidad social de poder comunicarse. Las nuevas Tecnologías de la Información y las Comunicaciones, conocidas como TICs, han provocado una serie de cambios sociales, económicos y políticos en todos los sectores de la sociedad.

Los nuevos tiempos han generado nuevos actores: Internet, la telefonía móvil, los videojuegos y demás artefactos digitales, que están cambiando la experiencia en múltiples aspectos: en el entretenimiento, en las comunicaciones personales, en el aprendizaje y en el trabajo. Internet ha alterado las reglas de juego tradicionales de elaboración, distribución y consumo de la información. En Internet ya no solo se encuentran textos o documentos para leer, ahora y de forma más creciente es un territorio de expresión multimedia y audiovisual. Además brinda la posibilidad de realizar una comunicación simultánea entre varios usuarios, mediante mensajes, correos y videoconferencia a través de la telefonía móvil o la propia Internet.

Con la evolución de la Web o Internet los usuarios dejan de ser usuarios pasivos para convertirse en usuarios activos, que participan y contribuyen en el contenido que se publica, surge así la Web 2.0. Aparece también una variable fundamental en el control de los mensajes que viajan a través de la red: la interactividad. Con este término ya no solo se reciben los mensajes enviados por algún proveedor, sino que se cuenta con un canal de retorno para poder tener una retroalimentación con esos contenidos, dejando de ser un simple espectador. Una de las claves fundamentales de la interactividad es que permite que se puedan intercambiar los roles de transmisor y receptor, además, se incita a controlar el contenido.

La videoconferencia es una tecnología que proporciona interactividad al ofrecer una comunicación directa entre dos o más personas que se encuentran ubicadas en diferentes lugares, a través del envío y recepción de audio, video y datos. A nivel mundial, las videoconferencias han alcanzado un mayor desarrollo y su uso se hace más frecuente. Son utilizadas por empresas u organizaciones para la realización de reuniones virtuales e intercambios comerciales, aunque el mayor impacto de la misma se refleja en la esfera de la educación, con la promoción de la superación a distancia. Cuba no se encuentra ajena al uso de las videoconferencias ya que se experimentan soluciones en la esfera de la educación y la salud.

La Universidad de las Ciencias Informáticas es un pilar en el desarrollo de software en Cuba. La misma cuenta con el Centro de desarrollo de Geoinformática y Señales Digitales (GEYSED), el cual desarrolla varias aplicaciones que gestionan y utilizan audiovisuales dentro de sus procesos. Un ejemplo de ello lo constituye el Sistema de Transmisión de Canales Virtuales (STCV) que brinda soluciones para la monitorización de la transmisión de video a través de las redes de datos con el uso de la tecnología *streaming*¹. También posibilita la planificación y programación de carteleras y recursos que serán necesarios para la creación, transmisión y visualización de las señales televisivas. El STCV cuenta también con una Plataforma Web Interactiva que posibilita la ejecución de aplicaciones que proporcionan interactividad a las emisiones de archivos multimedia realizadas por parte del subsistema de transmisión.

Los usuarios o receptores de los canales televisivos que utilicen el STCV pueden conectarse a la Plataforma Web Interactiva utilizando una computadora o cualquier dispositivo inteligente² que permita realizar una conexión, desde las instalaciones que es desplegada dicha plataforma. La conexión a la plataforma actualmente no incluye una comunicación entre usuarios sin la necesidad de usar terceros. Estos problemas de comunicación retrasan la colaboración e impiden el intercambio de ideas, experiencias y conocimientos en los que se definen estrategias para solucionar problemas de conjunto. Además imposibilitan el uso de alternativas viables para promover las funcionalidades del propio sistema. La interactividad de dicha plataforma se ve limitada con la falta de comunicación existente entre los usuarios, comunicación que en muchas ocasiones se requiere sea de manera visual y directa. Dicha comunicación pudiera ser utilizada para el establecimiento de teleconsultas, videoconferencias y reuniones virtuales. Lo que permitirá mostrar gráficos u otros medios que no se encuentren de manera digital, lograr un mejor entendimiento y enriquecimiento de la comunicación establecida.

A partir de la situación problemática descrita, se plantea el siguiente **problema a resolver**: ¿Cómo garantizar la comunicación de forma visual entre usuarios de la Plataforma Web Interactiva del Sistema de Transmisión de Canales Virtuales? El **objeto de estudio** de la investigación es la transmisión de

¹ Tecnología que permite distribuir contenido multimedia desde un ordenador servidor, a múltiples clientes.

² Son aquellos dispositivos electrónicos, que presentan un sistema que le permite relacionarse e intercambiar información con otros dispositivos.

videoconferencias sobre la web, enmarcado en el **campo de acción**, el proceso de transmisión de videoconferencia en la Plataforma Web Interactiva del Sistema de Transmisión de Canales Virtuales. Se define como **objetivo general**: Desarrollar una Aplicación web de videoconferencia para la Plataforma Web Interactiva que garantice la comunicación visual entre usuarios.

El objetivo propuesto indujo a formular, como guías para el desarrollo de la investigación, las siguientes **preguntas científicas**:

1. ¿Cuáles son los referentes teóricos para el desarrollo de una aplicación web de videoconferencia para la Plataforma Web Interactiva?
2. ¿Qué herramientas y tecnologías son apropiadas para la transmisión de la videoconferencia dentro de la Plataforma Web Interactiva?
3. ¿Cómo estructurar el proceso de desarrollo de la aplicación web de videoconferencia que permita establecer una comunicación visual entre usuarios?
4. ¿El uso de la aplicación web de videoconferencia en la Plataforma Web Interactiva garantiza la comunicación visual entre usuarios?

Para dar cumplimiento al objetivo general se definen las siguientes **tareas de la investigación**:

1. Definición de los principales conceptos, características y tendencias actuales relacionadas con la videoconferencia en la web.
2. Análisis del estado actual de la Plataforma Web Interactiva, del Sistema de Transmisión de Canales Virtuales. Sus aplicaciones, componentes y necesidades.
3. Caracterización de las soluciones similares existentes en la Universidad de las Ciencias Informáticas, a nivel nacional e internacional.
4. Selección de la metodología de desarrollo, las herramientas y tecnologías a utilizar para la construcción de la aplicación web de videoconferencia
5. Implementación de la aplicación web de videoconferencia para la Plataforma Web Interactiva.
6. Realización de pruebas a la aplicación web de videoconferencia para validar su correcto funcionamiento y su integración con la Plataforma Web Interactiva.

Para la investigación se han utilizado diferentes tipos de **métodos de investigación**, tanto teóricos como empíricos, los cuales se describen a continuación:

Métodos Teóricos

- **Histórico-Lógico:** El método histórico estudia la trayectoria real de los fenómenos y acontecimientos en el transcurso de su historia. El método lógico investiga las leyes generales del funcionamiento y desarrollo de los fenómenos (Alvarez, 1995). Este método se utiliza para estudiar la evolución de los conceptos asociados a la videoconferencia y el proceso de transmisión de la misma sobre la web, permitiendo la definición de términos propios.
- **Analítico-Sintético:** Permite la descomposición de un todo complejo en sus partes y cualidades. La síntesis, por su parte, establece la unión entre las partes, previamente analizadas y posibilita descubrir relaciones y características generales entre los elementos de la realidad (Alvarez, 1995). Este método se utiliza para seleccionar las herramientas y tecnologías a utilizar durante el desarrollo de la aplicación, además permite la evaluación de otras soluciones que respondan al problema, lo que permite realizar una valoración crítica y detallada de cada una de ellas.

Métodos Empíricos

- **La entrevista:** es una técnica de recopilación de información mediante una conversación profesional, con la que además de adquirirse información acerca de lo que se investiga, tiene importancia educativa; y depende en gran medida del nivel de comunicación entre el investigador y los participantes en la misma (Álvarez, 1995). Se hace uso del método aplicado a los especialistas y directivos del departamento de Señales Digitales del Centro GEYSED con el fin de comprender el proceso de funcionamiento del STCV y describir e identificar las nuevas necesidades a solucionar.

Estructura del documento

La presente investigación consta de tres capítulos. El primero titulado “Fundamentos teóricos sobre la transmisión de videoconferencias en la web” donde se realiza la elaboración del marco teórico. Además, se efectúa una evaluación de los resultados obtenidos en trabajos precedentes y se definen las herramientas y lenguaje de programación a utilizar, así como la metodología de desarrollo más

recomendable. Seguidamente en el capítulo 2 que lleva por título “Análisis y diseño de *Fabulari*.”, se exponen las principales características y cualidades de la solución a implementar. Además se identifican los requerimientos, se escogen la arquitectura y los patrones de diseño, se diseñan las clases, y se detallan los pasos de la metodología propuesta, incluyendo varios de sus artefactos y diagramas. En el capítulo 3 titulado “Implementación y Pruebas de *Fabulari*”, se muestra la situación física de los distintos componentes lógicos desarrollados a través del modelo de despliegue y la organización del sistema mediante el modelo de componentes. Se explican los estilos de programación y estándares de codificación empleados. Y por último se valida el sistema desarrollado aplicándole las pruebas necesarias para demostrar que la solución es correcta.

Capítulo 1. Fundamentos teóricos sobre la transmisión de videoconferencias en la web

En el presente capítulo se exponen los aspectos teóricos sobre la transmisión de videoconferencias en la web. También se abordan aspectos importantes de soluciones existentes a nivel nacional e internacional, que se pudieran tener en cuenta para un mejor desarrollo de la aplicación. Y por último se describen y caracterizan la metodología, las herramientas y los lenguajes utilizados en la investigación.

1.1 Elementos asociados al dominio del problema.

A continuación se relacionan los principales conceptos o temáticas que están asociados al desarrollo de la investigación.

1.1.1 Aplicación web

Una aplicación web es una aplicación informática distribuida cuya interfaz de usuario es accesible desde un cliente web, normalmente un navegador web. El objetivo de los sistemas informáticos distribuidos es conectar a usuarios y recursos de manera transparente, abierta, rentable, fiable y adaptable. Los recursos físicos se pueden compartir, al igual que los virtuales, tales como sistemas operativos, tareas y aplicaciones (UIT, 2014). Para la actual investigación los autores se acogen a la definición brindada por (Pardo, 2006) en la que se plantea: “un aplicación web es la herramienta que aporta servicios de valor agregado y brinda un producto en evolución constante”.

1.1.2 Streaming

La necesidad de establecer un canal de datos entre computadores, por el cual se podrá distribuir contenido multimedia desde un ordenador servidor, a múltiples clientes, indica el uso de la tecnología *streaming* (Acevedo, y otros, 2015). El uso de *streaming* les permite a los usuarios establecer una comunicación a través de la videoconferencia y compartir video y audio para poder transmitir contenidos de interés entre ellos. Además les permite a dichos usuarios visualizarse de forma permanente en la videoconferencia sin la necesidad de grabar lo que se quiere transmitir y luego compartirlo para que otros usuarios puedan verlo.

1.1.3 Videoconferencia

El termino videoconferencia es una palabra compuesta que abarca dos conceptos dentro de la misma, video y conferencia. Desde su origen en latín (video) significa yo veo (videre: "ver" y eo: "ego"), el video

permite distribuir la señal visual a través de ondas y, particularmente, de impulsos eléctricos (López, 2010). Conferencia tiene asociados diferentes significados en dependencia del campo en el que se esté tratando. La conferencia es un tipo de exposición oral, impartida por especialistas, centrada en la presentación de un tema específico y de interés para el público al cual está destinada (Ruiz, 2011).

La videoconferencia es la *“Comunicación entre dos interlocutores que se encuentran en lugares distintos a través de una red de comunicaciones, un ordenador y una cámara de vídeo, de forma que, además de hablar, pueden verse”* (Larousse, 2015). Según (Acosta , y otros, 2015) la videoconferencia es el conjunto de hardware y software que permite la conexión simultánea por medio de audio y video para el intercambio de información interactiva, en tiempo real, entre personas que se encuentran geográficamente distantes. Por otra parte (Ribas, 2012) plantea que *“La videoconferencia es un sistema de comunicación que permite mantener reuniones colectivas entre varias personas que se encuentran en lugares distantes. Esta comunicación se realiza en tiempo real, vía telefónica, y se transmite tanto la imagen como el sonido, en ambos sentidos.”*

Los conceptos vistos no comtemplan algunos aspectos con relación al tema tratado para la presente investigación, por lo que a partir de ellos los autores definen un nuevo concepto enriquecido por los estudiados y que se adapta al marco del problema. Se define como videoconferencia al *“sistema en el que intervienen elementos de hardware y software que proporciona una comunicación bidireccional de audio y video entre receptores y emisores por medio de un canal de comunicación”*.

1.2 Elementos fundamentales de las videoconferencias

Los sistemas de videoconferencia necesitan de elementos fundamentales para su funcionamiento, de aquí que varios autores como (Suarez, 2010) y (Díaz, y otros, 2000) coinciden en que para lograr una mejor visión del sistema de videoconferencia, se hace necesario tomar como punto de partida que: *“los sistemas de videoconferencia suelen subdividirse en tres elementos básicos que son: La red de comunicaciones, la sala de videoconferencia y el codec”*.

1.2.1 Las Redes de comunicaciones

Para la transmisión de un sistema de videoconferencia es necesario la existencia de un canal para la comunicación. En el ámbito digital y de Internet la vía de comunicación entre uno o más personas es a través de las redes de comunicación. Esta red debe permitir una comunicación bidireccional entre los puntos o nodos a conectar (González, 2010). En la presente investigación los autores utilizan la definición

dada por (Halsall, 2005) en la que se plantea que una red de comunicación es la conexión de diferentes computadoras que pueden comunicarse e intercambiar información con el uso de sus propios recursos.

Estas redes pueden ser clasificada en base a distintos aspectos como por ejemplo: alcance, tipo de conexión, tecnología, entre otros. En el ámbito de la presente investigación se hace uso de las redes de comunicación por alcance, dentro de las cuales se puede mencionar las Redes Inalámbricas de Área personal (WPAN, por sus siglas en inglés), las Redes de Área Local (LAN, por sus siglas en inglés), las Redes Inalámbricas de Área Local (WLAN, por sus siglas en inglés), entre otras. En estas redes los dispositivos interconectados presentan una relación cliente/servidor en la que básicamente un cliente realiza peticiones desde un dispositivo a otro (servidor), quien le envía una respuesta al cliente atendiendo a determinada solicitud (Halsall, 2005).

1.2.2 La sala virtual de videoconferencia

Al referirse a una sala virtual se plantea que es un espacio empleado para realizar encuentros entre dos o más personas en distintos lugares físicos, utilizando Internet como medio de conexión (Libreeducación, 2013). Por otra parte (Manolakis, y otros, 2015) hace mención a la sala virtual asíncrona como forma de espacio digital que permite acceder y compartir recursos como texto, audio y video. Al carecer de un concepto completo, que se adecue al objetivo perseguido, los autores de la presente investigación definen que una sala virtual de videoconferencia será: *“el espacio digital para poder establecer una comunicación entre dos o más personas, en la que se podrá compartir recursos como texto, audio, video y datos, mediante el uso de una red de comunicación”*.

Para el establecimiento de una videoconferencia son necesarios algunos equipos que garanticen la visualización, la audición y la captura de las imágenes que serán transmitidas, entre estos se encuentran:

Monitor/ Pantalla: es una unidad de visualización de un equipo, por la que se visualiza la imagen local que se transmite. Los monitores pueden ser de diversos formatos en dependencia de las necesidades del usuario, pueden tener disímiles medidas (Ribas, 2009).

Cámara: Son las utilizadas para llevar a cabo la videoconferencia, es decir, las que van a captar la imagen de los participantes para transmitirla al otro extremo. Las cámaras pueden ser fijas o motorizadas. También se utilizan cámaras de documentos para la visualización de documentos escritos, gráficos, diapositivas y elementos sólidos (Ribas, 2009).

Micrófono: Son los encargados de capturar la señal sonora. Pueden ser de sobremesa, de mano o inalámbrico (Ribas, 2009).

1.2.3 El Codec

Otro elemento fundamental de la videoconferencia es el codec, el cual se encarga de ser Codificador/Decodificador de la información transmitida a través de la red. Algunos de los códec usados en el mundo de la videoconferencia en el video son: H.261, H.263, H.263+, H.263++, H.264 y en el audio: G.711, G.722, G.723, G.728, G.729 (González, 2010). Para la presente investigación se hace uso de la API WebRTC para el establecimiento de la comunicación, la cual presenta una licencia libre, Distribución de Software *Berkeley* (BSD por sus siglas en inglés). Para la codificación de voz WebRTC utiliza como codificación G.711 el cual ofrece una mejor calidad de audio y utiliza un ancho de banda de 64 Kbps, el cual no necesita licencia para su uso. Por otro lado para la codificación de video utiliza VP8, aunque (Millán, 2014) menciona que para un futuro utilizará la codificación H.264.

1.3 Estudio de soluciones existentes

A continuación se expone el estudio realizado sobre diferentes aplicaciones que tienen entre sus funcionalidades la transmisión de videoconferencias en la web. Se caracterizan productos nacionales y otros internacionales, principalmente aquellos que han sido desarrollados por empresas líderes en el trabajo con audiovisuales.

GEVITR: La UCI desarrolló en el año 2013 una herramienta para la gestión de videoconferencia interactivas en tiempo real con el objetivo de permitir la interacción entre estudiantes y profesores durante la participación en un aula virtual. La misma permite adicionar, listar y visualizar conferencia, subir y descargar archivo, entre otras características. Dicha herramienta hace uso de un servidor Web Apache y otro de flujo de video *Flumotion* v0.10 (Pérez, 2013) utilizando tecnologías libres. Es multiplataforma ya que es accesible desde varios Sistemas Operativos como Windows y GNU/Linux con el uso de un navegador web. La comunicación en esta herramienta es síncrona, por lo que si alguna de las peticiones que recibe tiene algún problema, puede colapsar dejando colgadas las demás peticiones. Al estar basado en tecnología PHP, cada conexión consume una pequeña parte de los recursos del servidor, ello implica que si producen un número elevado de peticiones recurrentes se consuman todos los recursos del servidor, llevando al colapso al mismo. Además le falta mejorar su interfaz para hacerla más sencilla y accesible desde otros dispositivos.

Marte 3.0 es un sistema multiplataforma de videoconferencia y compartición de aplicaciones basado en el Protocolo de Inicio de Sesión (SIP, por sus siglas en inglés), de código privativo. Las conferencias

soportadas por Marte permiten a los usuarios cambiar el modo en el que se le presentan los medios que recibe del servidor; si alguno cambia el modo de interacción, los demás reciben el mismo cambio, de forma que el esquema mostrado para todos ellos es el mismo. El servidor es flexible en cuanto a dichas configuraciones, permite mostrar el vídeo de un único participante, el vídeo de varios simultáneamente (otorgando prioridad a uno si así se desea), y compartir el escritorio de uno o varios participantes. Martes 3.0 utiliza un servidor Flash el cual influye en la calidad de la imagen al transmitir la videoconferencia, ya que el navegador tarda más en cargar una página con *Flash*. Para la visualización se necesita tener instalado el *Plugins* de *Flash* en el navegador lo que implicaría depender del uso de terceros para lograr la transmisión de la videoconferencia (Cerviño, y otros, 2008).

Skype es un software multiplataforma que permite hacer llamadas telefónicas usando una computadora o teléfono inteligente y una conexión a Internet. Utiliza un protocolo de Internet del tipo voz sobre IP, también conocido como VoIP, el cual hace posible que las señales de voz y vídeo sean transformadas en paquetes digitales y enviados a través de Internet. Aunque existen varias aplicaciones compatibles con Skype accesibles desde Sistemas Operativos como Windows, GNU/Linux o Android, pero las mismas no son funcionales al 100% ya que algunas características se encuentran limitadas, como por ejemplo las videoconferencias grupales, por ser un software privativo (Velasco, 2015).

RECOL es un sistema de videoconferencia a través de Internet que permite la conexión entre distintos usuarios a través de computadoras y teléfonos móviles. Su uso se basa en el pago de licencias de uso (por horas, días o semanas). Es una aplicación informática que funciona sobre plataforma UNIX, que permite la videoconferencia entre diversos usuarios simultáneos, así como compartir aplicaciones entre ellos, enviar ficheros, navegar conjuntamente por Internet o participar en un chat general o selectivo (Recol, 2007).

A continuación se muestra una tabla sinóptica que permite una rápida comprensión de las principales características identificadas por cada aplicación analizada y otras de interés propio de la investigación. Las casillas marcadas con una "x" indican la presencia de una determinada característica y, las que están en blanco, su ausencia.

Tabla 1. Tabla sinóptica de estudio de homólogos

Sistemas Detectados	Características de interés	Multiplataforma	Integración en cuanto a la autenticación de usuarios.	Gestión de Sala	Compartición de archivos	Código abierto
GEVTR		X		X	X	X
Marte 3.0		x			x	
Skype		X				
RECOL					X	

Después de analizadas las aplicaciones anteriormente descritas se concluye que las mismas no constituyen una solución para la problemática planteada en la presente investigación. **GEVTR** impide realizar una comunicación asíncrona y al estar basado en PHP puede producir un colapso del servidor ante un número elevado de peticiones recurrentes. **Marte 3.0** al ser un sistema de código privativo y utilizar un servidor *Flash* que relentiza la videoconferencia al transmitirla por medio de un navegador. **Skype** es un software privativo que presenta limitantes ya que no se puede realizar videoconferencias de forma libre. **RECOL** es un sistema privativo solamente funcional sobre la plataforma UNIX lo que limita su accesibilidad. Ante lo planteado con anterioridad será necesario desarrollar una aplicación web de videoconferencia haciendo uso de tecnologías libres que realice una comunicación asíncrona, permita gestionar salas, compartir archivos, sea multiplataforma y permita una integración con la Plataforma Web Interactiva en cuanto a la autenticación de sus usuarios para dar solución al problema de la investigación. Vale destacar que las soluciones estudiadas aportaron características a tener en cuenta para el desarrollo de la propuesta de solución.

1.4 Resultados obtenidos en el trabajo precedente.

La Plataforma Web Interactiva a través de las aplicaciones que tiene integradas permite crear un ambiente interactivo entre los usuarios y dichas aplicaciones, orientadas principalmente a la televisión, pero

actualmente no ofrece una alternativa que permita la comunicación de manera visual y directa entre varios usuarios ubicados en diferentes lugares. Ante la problemática se decide desarrollar una aplicación web de videoconferencia que pueda integrarse a la Plataforma Web Interactiva y que presente características como la autenticación de los usuarios de la plataforma, la gestión de usuarios y videoconferencias, la opción de compartir archivos, la administración de salas de videoconferencia y el uso de un chat. Con la aplicación los usuarios de la Plataforma Web Interactiva accederían por medio de dispositivos inteligentes y lograrían comunicarse de forma directa y visual.

Los humanos se comunican mediante palabras, etiquetan con ellas cualquier elemento tangible o abstracto de la realidad, ya sea un producto, un servicio, una empresa, o cualquier otra iniciativa. Por lo que es necesario etiquetarla con un nombre, para cuando las personas lo escuchen o lean automáticamente piensen en el elemento al que identifican. A partir de las premisas planteadas se decide nombrar a la aplicación web de videoconferencia a desarrollar, *Fabulari*. Este término proviene del latín y significa hablar, conversar o charlar de forma familiar. Se escoge este término pues resulta fácil de memorizar, pronunciar y su significado va muy acorde al objetivo fundamental que se persigue con la aplicación a desarrollar, que es precisamente lograr la comunicación a través de una conversación en una videoconferencia.

1.5 Metodología de desarrollo. Proceso Unificado de Desarrollo

Las Metodologías de desarrollo son un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevos software. Según (Kruchten, 2003) "*Las metodologías de desarrollo de software definen quién está haciendo qué, cuándo y cómo para alcanzar un determinado objetivo.*" Son usadas para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.

Para guiar el proceso de desarrollo de software de la presente investigación se utilizó como metodología el Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés) ya que sigue la línea de desarrollo del STCV. Al mantener RUP como metodología robusta y precisa, permite controlar y documentar el desarrollo del software, eliminando los riesgos que puedan existir en el proceso, propiciando en todo momento un enfoque de trabajo bien estructurado. Así como la asignación correcta y óptima de tareas y responsabilidades, lo que garantiza que los documentos ingenieriles (artefactos) desarrollados estén en correspondencia con los utilizados en el STCV.

El ciclo de vida de RUP según (Kruchten, 2003) se caracteriza por ser:

Dirigido por casos de uso: permite que los CU reflejen lo que el usuario necesita y desea, lo cual se representa a través de los requisitos. A partir de aquí los CU guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los CU.

Centrado en la arquitectura: muestra una visión común del sistema completo, por lo que se describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.

Iterativo e Incremental: permite que los flujos de trabajo se desarrollen en cascada, que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requisitos y obtiene un producto con un determinado nivel, incrementándose en cada iteración.

1.6 Herramienta CASE de modelado Visual Paradigm for UML 8.0

Para facilitar la realización de prototipos, el desarrollo conjunto de aplicaciones, mejorar y estandarizar la documentación se hace uso de una herramienta de Ingeniería de Software Asistida por Computadora (CASE, por sus siglas en inglés). Se considera una herramienta CASE a un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de software, completamente o en alguna de sus fases (UCOL, 2006). Además aumentan la portabilidad de las aplicaciones, facilitan la reutilización de componentes y posibilitan un desarrollo y refinamiento visual de las aplicaciones, mediante la utilización de gráficos. Al utilizar una herramienta CASE en el desarrollo de un proyecto se garantiza un aumento en la calidad y por consiguiente, un aumento de la productividad (INEI, 1999). Para el desarrollo de la solución se utilizó *Visual Paradigm for UML 8.0*.

Visual Paradigm es una herramienta multiplataforma, característica que la favorece y que viene muy acorde a la migración al software libre que lleva a cabo Cuba. Es una herramienta UML profesional que soporta el ciclo de vida del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Se integra con varios IDEs (Entorno de Desarrollo Integrado) y soporta múltiples usuarios trabajando sobre un mismo proyecto. Ofrece interoperabilidad entre diagramas, ya que permite a

partir de un diagrama obtener otro que guarde relación con el mismo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (Visual Paradigm, 2014).

1.7 Lenguaje de Modelado Unificado (UML)

Un lenguaje de modelado es un conjunto de símbolos y reglas que están estandarizados y se utilizan para modelar parte de un diseño de software orientado a objetos. Comúnmente son utilizados en combinación con una metodología de desarrollo de software para llegar de una especificación inicial a la implementación (Pressman, 2010).

Autores como (Kruchten, 2003) definen a UML como un “Lenguaje cuyo vocabulario y reglas se centran en la representación conceptual y física de un sistema”. UML es un lenguaje que permite visualizar, especificar, construir y documentar sistemas independientemente de la metodología de análisis y diseño, pero siempre con una perspectiva orientada a objetos. UML sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas de software como para la arquitectura hardware donde se ejecuten. Cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de lo que se quiere representar. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. UML ofrece un estándar para describir un plano del sistema, con la inclusión de aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

La utilización de UML para la presente investigación brinda a los desarrolladores proveer de un vocabulario y reglas que permiten una comunicación estándar y robusta, ello permite una comprensión sencilla de los diagramas y conceptos que serán modelados. Se utiliza la versión 2.2 de UML debido a que es la utilizada por Visual Paradigm for UML 8.0 (López, 2011).

1.8 Entorno de Desarrollo Integrado PhpStorm

Un Entorno de Desarrollo Integrado (IDE por sus siglas en inglés) es un programa informático compuesto por un conjunto de herramientas de programación, díganse un editor de código, un compilador, un depurador y un constructor de Interfaz Gráfica de Usuario (GUI por sus siglas en inglés). Según (Skvorc, 2015) en la encuesta publicada “*Best PHP IDE in 2014 – Survey Results*” menciona los mejores IDE de desarrollo que utilizan comunidades que desarrollan PHP y brinda como resultado que el IDE más usado tanto para desarrollo de proyectos personales como empresariales es *PhpStorm*.

Se decide usar el IDE PhpStorm ya que provee de un completamiento inteligente de código, detecta código duplicado, mezcla lenguajes (JavaScript, SQL, XML), presenta navegación rápida y chequeo de errores al momento, que ayudaran en el desarrollo de la aplicación. Además es un editor de JavaScript avanzado basado en DOM, que posee navegación de código y *debugger* de *JavaScript*, que facilita la rápida detección de errores en el código. Es un IDE ligero, de fácil instalación, multiplataforma y de código abierto (JetBrains, 2015).

1.9 Marco de trabajo MEAN

Con el objetivo de llevar a cabo el desarrollo de *Fabulari* de forma organizada se hace uso de un marco de trabajo (*framework*, en inglés), el cual brinda una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos (Riehle, 2000). El marco de trabajo seleccionado es MEAN por simplificar y acelerar el desarrollo de aplicaciones web. El mismo combina el uso de 4 tecnologías que utilizan un lenguaje de programación en común (JavaScript), el gestor de base de datos MongoDB, los marcos de trabajo Express.js y Angular.js, y el servidor web Node.js. El empleo del mismo lenguaje de programación en todas las partes de la aplicación brinda al programador la posibilidad de manejar todos los ámbitos de la aplicación web, aunque se especialice en uno de ellos (MEAN.IO, 2015). A continuación se describen los principales beneficios y características de cada una de las tecnologías que conforman el marco de trabajo MEAN para el desarrollo de *Fabulari*:

MongoDB es un sistema de base de datos NoSQL de código abierto, con su uso se podrá alcanzar un alto rendimiento y disponibilidad de la información que brindará *Fabulari*, además brinda una base de datos escalable. Dentro de sus ventajas proporciona índices secundarios, consultas dinámicas y sus agregaciones no poseen alta complejidad. MongoDB mantiene los datos relacionales juntos en los documentos, ello proporciona rapidez en cuanto a la respuesta de las consultas. Al emplear MongoDB con Node.JS, aporta más velocidad al servidor web cuando se ejecutan consultas sobre la base de datos (MongoDB, 2015).

Express es un marco de trabajo robusto, rápido, flexible y muy simple. Ofrece un mecanismo que combina una dirección, un método de solicitud HTTP (GET, POST, y entre otros), y uno o más controladores, con el fin de realizar peticiones de conexión a la Base de Datos MongoDB y definir rutas para el direccionamiento dentro de la aplicación (Express, 2015).

Angular es un marco de trabajo JavaScript de código abierto desarrollado por Google con el cual se pueden crear aplicaciones web basadas en JavaScript Asíncrono y XML (AJAX por sus siglas en inglés).

Con AngularJS se minimiza la complejidad de crecimiento de la aplicación a gran escala, de forma que permite ofrecer una estructura de desarrollo a *Fabulari*. También permite crear una aplicación con un código limpio y de buen entendimiento con el uso del patrón Modelo-Vista-Controlador (MVC) (Amazon, 2015).

NodeJs hace un cambio en el paradigma de la programación ya que lleva *JavaScript* fuera del navegador web y lo ejecuta del lado del servidor. Está basado en eventos y construido sobre el motor de *JavaScript* v8, uno de los intérpretes más rápidos para cualquier lenguaje dinámico, además de tener un respaldo de su comunidad que realiza documentos, tutoriales y crea nuevos módulos (NodeJS, 2014). Una de las principales ventajas de NodeJs es que posee un modelo Entrada/Salida que permite atender un gran número de clientes a la vez, ello ayudará a mantener una comunicación asíncrona en el servidor (Muñoz, 2013).

1.10 Biblioteca JQuery

JQuery es utilizado en el desarrollo de *Fabulari* ya que permite el manejo de eventos, la animación, y las interacciones Ajax para un desarrollo web rápido. Utiliza JavaScript para interactuar con HTML y CSS. Sus selectores son simples, por lo que es una biblioteca fácil de aprender para los diseñadores y desarrolladores por igual. Además posee una amplia comunidad de desarrollo en la que se comparten experiencias, así como *plugins* que complementan el trabajo, lo que le brinda a los programadores la facilidad de poder buscar una respuesta a posibles problemas que pudieran aparecer durante el desarrollo de *Fabulari* (Alvarez, 2012).

1.11 API WebRTC

WebRTC es un acrónimo de *Web Real-Time Communications* que en español significa comunicación en tiempo real a través de la web. WebRTC forma parte de HTML5 y permite establecer comunicaciones en tiempo real (llamadas de voz y vídeo) sobre los navegadores Web, ello brinda la posibilidad de no instalar extensiones no estándares o *plugins* propietarios. Los componentes WebRTC al estar implementados en el navegador pueden ser fácilmente accesibles mediante una API *JavaScript*, lo que permite poder integrar comunicaciones en tiempo real dentro de la aplicación. Sus beneficios se encuentran en la forma de comunicarse, ya que los medios en WebRTC van de navegador a navegador directamente, lo que reduce la latencia de la red y emplea más eficientemente el ancho de banda (Tejedor, 2014).

1.12 Lenguajes de programación

Un lenguaje de programación es cualquier lenguaje artificial, el cual, se utiliza para definir adecuadamente una secuencia de instrucciones que puedan ser interpretadas y ejecutadas en una computadora (Rena, 2015). Con el uso del marco de trabajo que se propone para el desarrollo de la solución (MEAN) se utilizará como lenguaje de programación *JavaScript*, ya que es el lenguaje nativo del mismo. También se hará uso del Lenguaje Marcado de Hipertexto versión 5 (HTML5 por sus siglas en inglés) y de las hojas de estilo en cascada (CSS3 por sus siglas en inglés).

El uso del lenguaje ***JavaScript*** es básico en el trabajo con el marco de trabajo propuesto ya que es utilizado tanto del lado del cliente como del lado del servidor. Además le brinda al programador un lenguaje ligero e interpretado, orientado a objetos, basado en prototipos y soporta estilos de programación funcional. El mismo permite manejar objetos dentro de la página web y sobre ese objeto se pueden definir diferentes eventos, lo que facilita la programación de páginas interactivas (Mozilla Foundation, 2015).

Con el uso del lenguaje **HTML5** para el maquetado de *Fabulari* brinda la posibilidad de poder mostrar instrucciones especiales que son interpretadas por el navegador y como resultado poder mostrar textos, imágenes, vídeos y enlaces. Permite el uso de una gran variedad de etiquetas que son las encargadas de mantener la estructura e indicar la semántica de un documento, convirtiéndose en un lenguaje para presentación con soporte de aspectos visuales. Brinda un nuevo conjunto de etiquetas para dar estructura a un documento HTML, como la etiqueta “video” que permite mostrar los contenidos multimedia de la videoconferencia (Vega, y otros, 2011).

El lenguaje **CSS3** brinda la posibilidad de definir de manera eficiente las reglas y estilos de representación, con el fin de poder visualizar la aplicación en diferentes dispositivos, ya sean pantallas de equipos de escritorio, portátiles, móviles u otros dispositivos capaces de mostrar contenidos web. Brinda un control centralizado de la presentación de *Fabulari* lo que agiliza la actualización del mismo (HTML5, 2014).

1.13 Protocolo WebSocket

Para la comunicación se hará uso del protocolo *WebSockets* ya que es entendido por el servidor NodeJs, define un canal de comunicación bidireccional y *full-duplex*³ que opera a través de un *socket*⁴ sobre el Protocolo de Control de Transmisión (TCP por sus siglas en inglés) que posibilita intercambiar información por un mismo canal de comunicación. La utilización de un *socket* brinda una manera más sencilla de poder establecer la comunicación entre el cliente y servidor y reduce el tráfico innecesario sobre la red (jWebSockets, 2014).

1.14 Conclusiones parciales

El estudio de los principales conceptos relacionados a la videoconferencia y al proceso de transmisión de las mismas a través de la web permitió sentar las bases para el desarrollo de la propuesta de solución, además del porqué y la necesidad de la realización del presente trabajo. El estudio de sistemas que utilizan videoconferencias en el mundo evidenció que el uso de los mismos es necesario para los diferentes sectores de la sociedad, pero se comprobó que las soluciones existentes no constituyen una solución para la problemática de la investigación, de ahí el hecho de desarrollar una aplicación web de videoconferencia. La utilización del marco de trabajo MEAN, el IDE PhpStorm y los lenguajes de programación JavaScript, HTML5 y CSS3, además de Visual Paradigm como herramienta CASE facilitan y fortalecen el trabajo, aportan rapidez y eficacia a lo largo del ciclo de vida del desarrollo y garantizan una mayor usabilidad de la aplicación.

³ Una comunicación es *full-duplex* si puede enviar y recibir datos al mismo tiempo, lo que puede ocurrir entre dos nodos conectados directamente (Halsall, 2005).

⁴ Es un "canal de comunicación" entre dos programas que corren sobre ordenadores distintos o incluso en el mismo ordenador. Está conformado por un número de identificación compuesto por otros dos números: la dirección IP y el número de puerto TCP (Halsall, 2005).

Capítulo 2. Análisis y diseño de *Fabulari*.

En el presente capítulo se exponen las principales características y cualidades del sistema a implementar, mediante la identificación de los requisitos funcionales y no funcionales con los que debe de cumplir la aplicación que dará solución a la problemática de la presente investigación. También se describen los conceptos relacionados con las definiciones de estilo y patrones de diseño de software empleados. Además se detallan los pasos de la metodología que se propone en el capítulo anterior (RUP). Se incluye la elaboración del modelo de dominio, la definición de actores y relaciones entre ellos, además de los diagramas de casos de uso del sistema y las descripciones textuales de los casos de uso. Y por último se muestran los diagramas de clases del diseño propuestos.

2.1 Modelo de dominio.

El Modelo de Dominio o Modelo Conceptual es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Para la realización del mismo se utiliza un lenguaje común, lo que facilita la comunicación entre los desarrolladores y usuarios de la aplicación y un mayor entendimiento del contexto en que se desarrolla el sistema. El modelo de dominio se describe mediante diagramas UML (diagrama de clases) que muestra las clases del dominio y cómo se relacionan unas con otras mediante asociaciones (Larman, 2003). Para la presente investigación se plantea el uso de un modelo de dominio ya que no se tiene una estructura definida de los procesos del negocio, no se cuenta con un manual de procedimientos y los flujos de información son difusos.

2.1.1 Descripción general del Modelo de Dominio.

Los usuarios que pertenecen al producto Sistema de Transmisión de Canales Virtuales pueden acceder a la Plataforma Web Interactiva por medio de un Dispositivo Inteligente a través de una red de datos. Dicha plataforma contiene aplicaciones que proporcionan interactividad al STCV.

2.1.2 Descripción de las clases del Modelo de Dominio.

Como el modelo de dominio o modelo conceptual contribuye posteriormente en el proceso de desarrollo del software a identificar las clases que se utilizarán para modelar el sistema, a continuación se identifican los conceptos fundamentales que se emplean en el modelo a través de un glosario de términos:

Usuario: Son los usuarios autenticados en la Plataforma Web Interactiva, que acceden a la misma a través de los dispositivos.

Dispositivos Inteligentes: Entiéndase aquellos dispositivos electrónicos, ya sean computadoras, tabletas electrónicas, teléfonos y televisores inteligentes, que presentan un sistema que le permite relacionarse e intercambiar información con otros dispositivos inteligentes a través de una red de datos.

Plataforma Web Interactiva: Es el sistema encargado de brindar una serie de servicios a los usuarios como subir archivos multimedia y aplicaciones las cuales serán instaladas.

Aplicaciones: Aplicaciones instaladas en la Plataforma Web Interactiva.

2.1.3 Diagrama de clases del Modelo de Dominio.

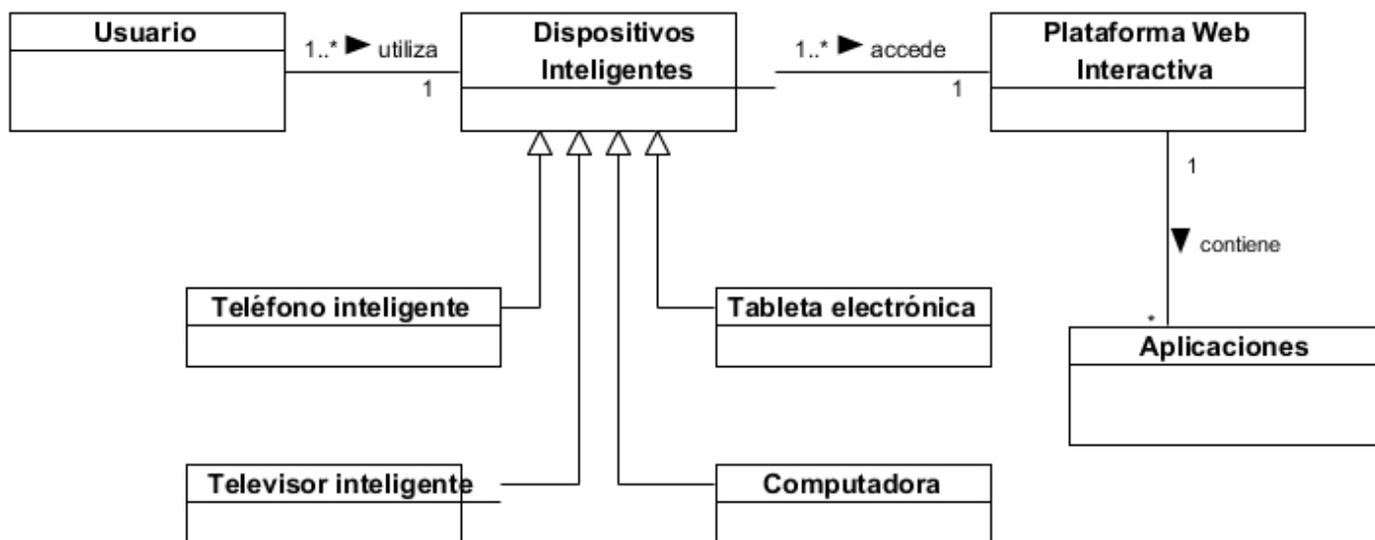


Figura 1. Representación del Modelo de Dominio

2.2 Especificación de los requisitos de la aplicación web de videoconferencia, *Fabulari*.

El Instituto de Ingeniería de Software (SEI, por sus siglas en inglés) plantea como parte del glosario del modelo CMMI que un requisito es: (1) Una condición o capacidad necesitada por un usuario para solucionar un problema o lograr un objetivo. (2) Una condición o capacidad que debe cumplir o poseer un producto o componente de producto para satisfacer un contrato, un estándar, una especificación u otros documentos impuestos formalmente. (3) Una representación documentada de una condición o capacidad como en (1) o en (2) (Ramos, 2013).

2.2.1 Requisitos funcionales.

Los requisitos funcionales (RF) son capacidades o condiciones que el sistema debe cumplir. Son establecidos entre el cliente y los desarrolladores sobre lo que debe o no hacer el sistema (Kruchten, 2003). Los requisitos identificados para el desarrollo de *Fabulari* son los siguientes:

RF1. Autenticar usuarios de la Plataforma Web Interactiva

La aplicación debe permitir autenticar directamente a los usuarios que accedan desde la Plataforma Web Interactiva.

RF2. Crear sala de videoconferencia

La aplicación debe permitir a los usuarios crear una sala de videoconferencia.

RF3. Listar salas de videoconferencia

La aplicación debe permitir listar las salas de videoconferencia creadas por el usuario y a las que ha sido invitado.

RF4. Editar sala de videoconferencia

La aplicación debe permitir editar los datos asociados a una sala de videoconferencia.

RF5. Buscar sala de videoconferencia

La aplicación debe permitir buscar una sala de videoconferencia, al introducir los datos en un criterio de búsqueda.

RF6. Eliminar sala de videoconferencia

La aplicación debe permitir eliminar una sala de videoconferencia.

RF7. Visualizar videoconferencia

La aplicación debe permitir visualizar la videoconferencia que el usuario ha creado o a la que ha sido invitado.

RF8. Visualizar usuarios conectados a la sala de videoconferencia

La aplicación debe permitir visualizar los usuarios que se encuentran conectados a una sala de videoconferencia.

RF9. Comunicar por chat

La aplicación debe permitir la comunicación entre usuarios mediante un chat.

RF10. Subir archivo a la sala de videoconferencia

La aplicación debe permitir subir archivos a la sala de videoconferencia.

RF11. Descargar archivo de la sala de videoconferencia

La aplicación debe permitir a los usuarios descargar o no el archivo que le ha sido enviado por otro usuario.

RF12. Cerrar transmisión de videoconferencia

La aplicación debe permitir cerrar la transmisión de la videoconferencia.

RF13. Monitorizar transmisión de videoconferencia

La aplicación debe permitir al administrador poder visualizar todas las transmisiones de videoconferencia.

RF14. Listar usuarios de la aplicación

La aplicación debe permitir listar los usuarios de la aplicación.

RF15. Buscar usuario de la aplicación

La aplicación debe permitir buscar usuarios de la aplicación.

RF16. Modificar el rol de usuario de la aplicación

La aplicación debe permitir modificar el rol de un usuario determinado en la aplicación.

RF17. Invitar usuarios a una sala de videoconferencia

La aplicación debe permitir al usuario invitar o no a otros usuarios a una sala de videoconferencia.

2.2.2 Requisitos no funcionales.

Los requisitos no funcionales (RNF) son propiedades o cualidades que el producto debe tener. Especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, seguridad y fiabilidad. Para determinar los requisitos no funcionales se tuvo en cuenta las clasificaciones dadas por (Kruchten, 2003) en el libro *Rational Unified Process*.

Requisito de interfaz:

RNF 1. Debe ser visualizados en diferentes tipos de dispositivos como computadoras, teléfonos inteligentes, tabletas electrónicas, televisores inteligentes o cualquier dispositivo electrónico que permita acceder a una red de datos.

Requisito físico:

- RNF 2. La PC servidor donde se instalará la aplicación deberá cumplir con los siguientes requisitos mínimos que se especifican a continuación: una RAM de 2 GB o superior, un mínimo de disco duro de 80 GB y procesador a 3.0 GHz o superior.
- RNF 3. La aplicación es accesible desde una computadora que cumpla con los requisitos mínimos que se especifican a continuación: una RAM de 2 GB o superior, un procesador de 1.5 GHz y un almacenamiento de 80 GB.
- RNF 4. La aplicación es accesible desde otros dispositivos inteligentes (que no sean computadoras) que cumplan con los requisitos mínimos que se especifican a continuación: una RAM de 1 GB o superior, un procesador de 1.5 GHz y un almacenamiento de 2 GB.
- RNF 5. Dispositivos inteligentes que soporten los estándares 802.11 b/g/n o 802.3 u.
- RNF 6. Los dispositivos deben poseer una cámara web con al menos 320 x 240 píxeles de resolución, y preferentemente bocinas o algún dispositivo de salida de audio y un micrófono o algún dispositivo de entrada de audio, de no poseer estos últimos solo se podrá visualizar el video.

Requisito de software:

- RNF 7. Para un buen desempeño de la aplicación se recomienda el uso de un Navegador que implementen la tecnología WebRTC (su funcionamiento ha sido comprobado en PC con Google Chrome v39 y Mozilla Firefox v38 y para dispositivos móviles con Google Chrome v41 y Mozilla Firefox v38).
- RNF 8. El servidor donde esté instalada la aplicación debe tener en ejecución un Servidor de Base de Datos MongoDB v2.6 o superior y un Servidor NodeJS v10.35 o superior.

2.3 Descripción de *Fabulari*.

Los diagramas de casos de uso tienen como principal ventaja la facilidad para interpretarlos, de ahí su utilización para comprender fácilmente los requisitos funcionales, los cuales definen a su vez las funcionalidades que realizará el sistema, funcionalidades que serán inicializadas siempre por uno de los actores que interactúan con el mismo (Kruchten, 2003) . A continuación se definen los actores y los casos de uso acompañados de sus descripciones.

2.3.1 Definición de los actores.

Se le llama actor a toda entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad. Esto incluye a los operadores humanos pero también incluye a todos los sistemas externos, además de entidades abstractas, como el tiempo (Kruchten, 2003). A continuación se describen los actores de la aplicación a desarrollar.

Tabla 2. Descripción de los actores del sistema.

Actor	Objetivos
Usuario	Es la persona que interactúa con la aplicación a través de la plataforma, tiene acceso a diferentes funcionalidades como la administración y visualización de videoconferencia.
Administrador	Se encarga de monitorizar y controlar todo lo que sucede en la aplicación de videoconferencia.
Sistema	Es el encargado de autenticar el usuario de la Plataforma Web Interactiva.

2.3.2 Diagrama de casos de uso del sistema.

Un diagrama de casos de uso explica gráficamente un conjunto de casos de uso (CU) del sistema, los actores y como se relacionan entre ellos. El diagrama tiene por objetivo ofrecer una clase de diagrama contextual que permite conocer rápidamente los actores externos de un sistema y las formas básicas en que lo utilizan (Larman, 2003).

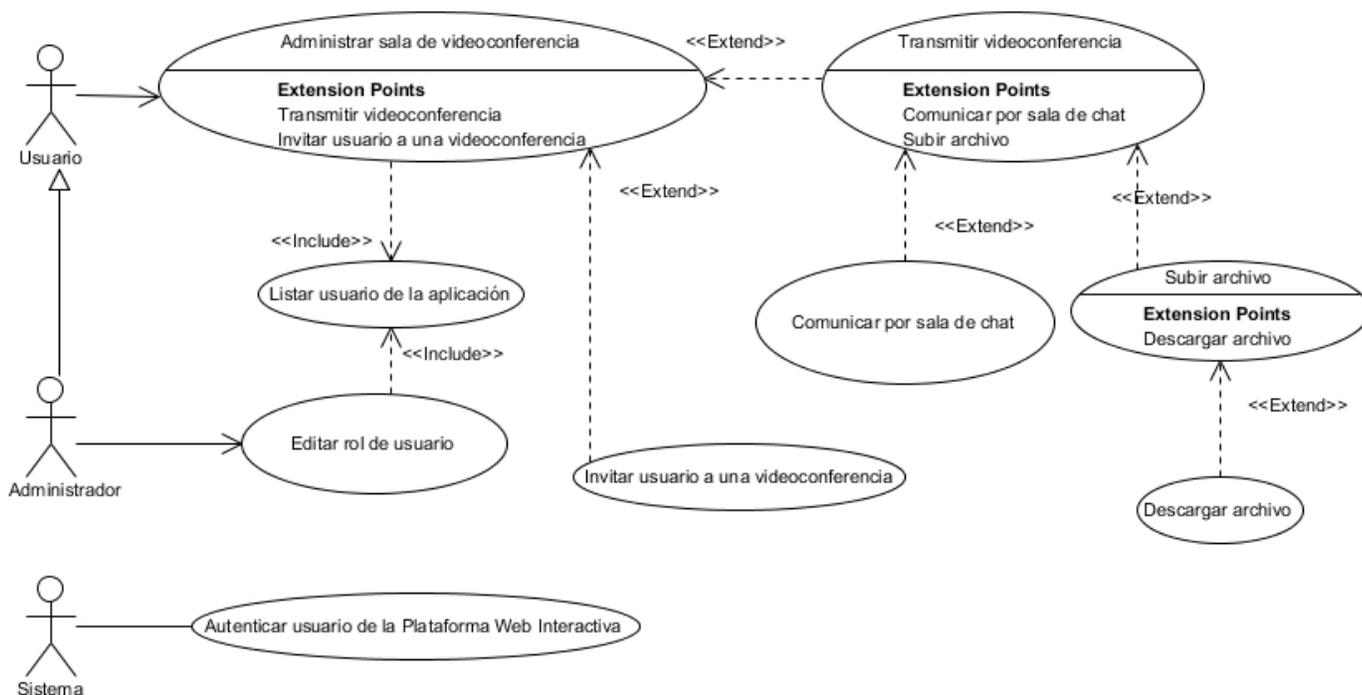


Figura 2. Diagrama de Caso de Uso.

2.3.3 Descripción de casos de uso.

Los CU dirigen el proceso de desarrollo en su totalidad. Son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario (Larman, 2003). A continuación se describe en la tabla 3 y tabla 4 los CU críticos de la aplicación web de videoconferencia *Fabulari*.

Tabla 3. Descripción de caso de uso: Administrar salas de videoconferencia.

Caso de Uso	Administrar salas de videoconferencia.
Objetivo	Listar, Crear, Editar, Eliminar o Buscar una sala de videoconferencia.
Actores	Usuario (Autenticado)
Resumen	El CU se inicia cuando el usuario ejecuta la aplicación, donde se listan las salas de videoconferencia creadas por el usuario y a las que él ha sido invitado. El mismo permite crear, editar, eliminar y buscar las salas de videoconferencia del usuario. El CU termina cuando el usuario sale de la aplicación o inicia una transmisión.
Complejidad	Alta
Prioridad	Crítico

Precondiciones	Que el usuario esté autenticado.	
Postcondiciones	Se Lista, Crea, Edita, Elimina o Busca una sala de videoconferencia.	
Flujo de eventos		
Flujo básico: Administrar sala de videoconferencia		
	Actor	Sistema
1.	Ejecuta la aplicación.	
2.		<p>Muestra una vista con una lista de las salas de videoconferencias creadas por el usuario y a las que él ha sido invitado, además muestra las siguientes opciones:</p> <ul style="list-style-type: none"> - Crear nueva Sala. Ver Sección 1: <i>“Crear nueva Sala”</i>. - Ver. Ver CU <i>“Transmitir videoconferencia”</i>. - Editar. Ver Sección 2: <i>“Editar sala de videoconferencia”</i>. - Eliminar. Ver Sección 3: <i>“Eliminar sala de videoconferencia”</i>. - Buscar. Ver Sección 4: <i>“Buscar sala de videoconferencia”</i>.
3.		Termina CU.
Sección 1: “Crear sala”.		
Flujo básico: Crear sala.		
	Actor	Sistema
1.	Presiona el botón “Crear nueva Sala”	

2.		Muestra una vista con los campos a llenar de la sala de videoconferencia. Inicia el Caso de Uso: Listar usuarios de la aplicación. Ver Caso de Uso " <i>Listar usuarios de la aplicación</i> ".
3.	Inserta los datos de la sala de videoconferencia: <ul style="list-style-type: none"> - Nombre - Fecha - Selecciona los usuarios que desea invitar a la sala. 	
4.	Presionar el botón " <i>Crear</i> ".	
5.		Verifica que los datos sean correctos y no existan campos vacíos.
6.		Crea la sala de videoconferencia y muestra una vista con los datos de la sala creada.
7.		Vuelve al paso 2 del Flujo Básico.
Flujos alternos		
4.1: El actor deja el campo "Nombre" vacío.		
	Actor	Sistema
4.1.1		Muestra un mensaje " <i>El nombre es requerido.</i> "
4.1.2		Vuelve a ejecutar el paso 2 de la Sección 1.
4.2 El actor llena el campo "Nombre" con más de 25 caracteres.		

	Actor	Sistema
4.2.1		Muestra un mensaje “ <i>La longitud máxima es de 25 caracteres.</i> ”
4.1.2		Vuelve a ejecutar el paso 2 de la Sección 1.
4.3 El actor llena el campo “Nombre” con caracteres extraños.		
	Actor	Sistema
4.3.1		Muestra un mensaje “ <i>No se admite caracteres extraños.</i> ”
4.3.2		Vuelve a ejecutar el paso 2 de la Sección 1.
4.4 El actor no especifica una fecha		
	Actor	Sistema
4.4.1		Muestra un mensaje “ <i>La fecha es requerida.</i> ”
4.4.2		Vuelve a ejecutar el paso 2 de la Sección 1.
Sección 2: “Editar sala de videoconferencia”.		
Flujo básico: Editar sala de videoconferencia.		
	Actor	Sistema
	Presiona el botón “Editar”.	
1.		Muestra la vista de la sala de videoconferencia seleccionada con los datos de la sala.
2.	Edita los datos de la sala de videoconferencia.	
3.	Presiona el botón “ <i>Guardar</i> ”.	

4.		Verifica que los datos sean correctos y no existan campos vacíos.
5.		Actualiza los datos de la sala de videoconferencia del usuario.
6.		Vuelve al paso 2 del Flujo Básico.
Flujos alternos		
4.1: El actor deja el campo "Nombre" vacío.		
	Actor	Sistema
4.1.1		Muestra un mensaje " <i>El nombre es requerido.</i> "
4.1.2		Vuelve a ejecutar el paso 2 de la Sección 2.
4.2 El actor llena el campo "Nombre" con más de 25 caracteres.		
	Actor	Sistema
4.2.1		Muestra un mensaje " <i>La longitud máxima es de 25 caracteres.</i> "
4.1.2		Vuelve a ejecutar el paso 2 de la Sección 2.
4.3 El actor llena el campo "Nombre" con caracteres extraños.		
	Actor	Sistema
4.3.1		Muestra un mensaje " <i>No se admite caracteres extraños.</i> "
4.3.2		Vuelve a ejecutar el paso 2 de la Sección 2
4.4 El actor no especifica una fecha		
	Actor	Sistema

4.4.1		Muestra un mensaje “ <i>La fecha es requerida.</i> ”
4.4.2		Vuelve a ejecutar el paso 2 de la Sección 2.
Sección 3: “Eliminar sala de videoconferencia”.		
Flujo básico: Eliminar sala de videoconferencia		
	Actor	Sistema
	El usuario presiona el botón “Eliminar”	
1.		Muestra un mensaje de confirmación ¿Seguro que desea eliminar la sala?
2.	Elige “Aceptar”	
		Elimina la sala de videoconferencia de la aplicación. Muestra un mensaje de información “Operación realizada. La sala ha sido eliminada”.
3.		Vuelve al paso 2 del Flujo Básico.
Sección 4: “Buscar sala de videoconferencia”.		
Flujo básico: Buscar sala de videoconferencia.		
	Actor	Sistema
1.	Se inserta el criterio de búsqueda en la opción buscar	
2.		Se muestra el listado con la información que cumpla con el criterio de búsqueda especificado.
3.		Vuelve al paso 2 del Flujo Básico.
Flujos Alternos		
1.1: Se inserta el criterio de búsqueda en la opción buscar y no se encuentra resultados.		
1.1.1		Si no existen resultados se muestra el siguiente mensaje: “ <i>No se encuentran resultados</i> ”

1.2	Vuelve a ejecutar el paso 1 de la Sección 4	
Relaciones	CU Incluidos	Listar Usuarios de la aplicación
	CU Extendidos	Invitar usuario a una videoconferencia en el CU Administrar sala de videoconferencia. Transmitir videoconferencia en el CU Administrar sala de videoconferencia.
Requisitos funcionales	no	<p>Requisito de interfaz:</p> <p>RNF 1. Debe ser visualizados en diferentes tipos de dispositivos como computadoras, teléfonos inteligentes, tabletas electrónicas, televisores inteligentes o cualquier dispositivo electrónico que permita acceder a una red de datos.</p> <p>Requisito físico:</p> <p>RNF 2. La PC servidor donde se instalará la aplicación deberá cumplir con los siguientes requisitos mínimos que se especifican a continuación: una RAM de 2 GB o superior, un mínimo de disco duro de 80 GB y procesador a 3.0 GHz o superior.</p> <p>RNF 3. La aplicación es accesible desde una computadora que cumpla con los requisitos mínimos que se especifican a continuación: una RAM de 2 GB o superior, un procesador de 1.5 GHz y un almacenamiento de 80 GB.</p> <p>RNF 4. La aplicación es accesible desde otros dispositivos inteligentes (que no sean computadoras) que cumplan con los requisitos mínimos que se especifican a continuación: una RAM de 1 GB o superior, un procesador de 1.5 GHz y un almacenamiento de 2 GB.</p> <p>RNF 5. Dispositivos inteligentes que soporten los estándares 802.11 b/g/n o 802.3 u.</p>

	<p>Requisito de software:</p> <p>RNF 6. Para un buen desempeño de la aplicación se recomienda el uso de un Navegador que implementen la tecnología WebRTC (su funcionamiento ha sido comprobado en PC con Google Chrome v39 y Mozilla Firefox v38 y para dispositivos móviles con Google Chrome v41 y Mozilla Firefox v38).</p> <p>RNF 7. El servidor donde esté instalada la aplicación debe tener en ejecución un Servidor de Base de Datos MongoDB v2.6 o superior y un Servidor NodeJS v10.35 o superior.</p>
Asuntos pendientes	No existe

Tabla 4. Descripción de caso de uso: Visualizar videoconferencia.

CU2. Visualizar videoconferencia.so de Uso	Transmitir videoconferencia
Objetivo	Visualizar usuarios conectados a la videoconferencia, visualizar videoconferencia, iniciar y cerrar la transmisión de la videoconferencia
Actores	Usuario (Autenticado)
Resumen	El CU inicia cuando el usuario selecciona el botón “Ver” en la vista Gestión de mis Salas y concluye cuando el usuario selecciona el botón “Salir” en la Sala de videoconferencia.
Complejidad	Alta
Prioridad	Crítico
Precondicions	La sala de videoconferencia ha sido creada y el usuario este autenticado.
Postcondicioes	Visualiza usuarios conectados a la videoconferencia, visualiza videoconferencia, inicia y cierra la transmisión de la videoconferencia

Flujo de eventos		
Flujo básico: Transmitir videoconferencia		
	Actor	Sistema
1.	Selecciona el botón "Ver"	
2.		<p>Muestra la vista de la sala de videoconferencia con los usuarios que están conectados a ella y permite:</p> <ul style="list-style-type: none"> - Comunicar por chat. Ver Caso de Uso 5: "<i>Comunicar por chat</i>". - Subir archivo. Ver Caso de Uso 6: "<i>Subir archivo a la sala de videoconferencia</i>". - Colgar transmisión de un usuario en la sala videoconferencia. Ver Sección 1: "<i>Colgar transmisión de un usuario en la sala videoconferencia</i>". - Salir de la sala de videoconferencia. Ver Sección 2: "<i>Salir de la sala de videoconferencia</i>".
Sección 1: "Colgar transmisión de un usuario en la sala videoconferencia".		
Flujo básico: Colgar transmisión de un usuario en la sala videoconferencia.		
	Actor	Sistema
	El usuario presiona el botón "Colgar"	
1.		Cierra la transmisión del usuario escogido en la sala de videoconferencia.

2.		Vuelve al paso 2 del Flujo Básico.
Sección 2: “Salir de la sala de videoconferencia”.		
Flujo básico: Salir de la sala de videoconferencia.		
	Actor	Sistema
	El usuario presiona el botón “Salir”	
1.		Saca de la sala de videoconferencia al usuario.
		Terminar CU.
Requisitos no funcionales	<p>Requisito de interfaz:</p> <p>RNF 1. Debe ser visualizados en diferentes tipos de dispositivos como computadoras, teléfonos inteligentes, tabletas electrónicas, televisores inteligentes o cualquier dispositivo electrónico que permita acceder a una red de datos.</p> <p>Requisito físico:</p> <p>RNF 2. La PC servidor donde se instalará la aplicación deberá cumplir con los siguientes requisitos mínimos que se especifican a continuación: una RAM de 2 GB o superior, un mínimo de disco duro de 80 GB y procesador a 3.0 GHz o superior.</p> <p>RNF 3. La aplicación es accesible desde una computadora que cumpla con los requisitos mínimos que se especifican a continuación: una RAM de 2 GB o superior, un procesador de 1.5 GHz y un almacenamiento de 80 GB.</p> <p>RNF 4. La aplicación es accesible desde otros dispositivos inteligentes (que no sean computadoras) que cumplan con los requisitos mínimos que se especifican a continuación: una RAM de 1 GB o superior, un procesador de 1.5 GHz y un almacenamiento de 2 GB.</p>	

	<p>RNF 5. Dispositivos inteligentes que soporten los estándares 802.11 b/g/n o 802.3 u.</p> <p>RNF 6. Los dispositivos deben poseer una cámara web con al menos 320 x 240 píxeles de resolución, y preferentemente bocinas o algún dispositivo de salida de audio y un micrófono o algún dispositivo de entrada de audio, de no poseer estos últimos solo se podrá visualizar el video.</p> <p>Requisito de software:</p> <p>RNF 7. Para un buen desempeño de la aplicación se recomienda el uso de un Navegador que implementen la tecnología WebRTC (su funcionamiento ha sido comprobado en PC con Google Chrome v39 y Mozilla Firefox v38 y para dispositivos móviles con Google Chrome v41 y Mozilla Firefox v38).</p>
Asuntos pendientes	No existe

2.4 Patrón arquitectónico

La arquitectura de software permite representar de forma concreta la estructura y funcionamiento interno de un sistema. Según (Clements, y otros, 2003): *“la arquitectura de software de un sistema es la estructura o estructuras del sistema, lo cual abarca componentes de software, las propiedades visibles externamente de esos componentes, y las relaciones entre ellas”*. Al diseñar una arquitectura de software se debe crear y representar componentes que interactúen entre ellos y tengan asignadas tareas específicas, además de organizarlos de forma tal que se logren los requisitos establecidos. Se puede iniciar con patrones de soluciones ya probados, y utilizar modelos que han funcionado. Estas soluciones probadas se conocen como patrones arquitectónicos, que van de lo general a lo particular.

Para la creación de *Fabulari* el patrón arquitectónico definido fue el Modelo Vista Controlador (MVC) ya que es el propuesto por el marco de trabajo MEAN, el cual es utilizado para el desarrollo de la aplicación. MVC separa los datos y la lógica de negocio de la aplicación, de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

Los tres componentes que conforman el patrón son el modelo, la vista y el controlador. El modelo administra el comportamiento y los datos del dominio de la aplicación, es un componente totalmente independiente de las representaciones específicas de salidas o del comportamiento de la entrada. La vista maneja la visualización de la información, implementa los procedimientos de actualización, además presenta la información del modelo al usuario. El controlador recibe la entrada, normalmente como eventos que codifican los movimientos del mouse o entrada del teclado, los eventos son traducidos para servir a las demandas del modelo o las vistas. El usuario interactúa con el sistema solamente a través de los controladores (Burbeck, 2002).

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo independientemente de la representación visual, lo que provoca como ventajas, sencillez para crear distintas representaciones de los mismos datos, permite la reutilización de los componentes, simplicidad en el mantenimiento de la aplicación y facilidad para desarrollar prototipos rápidos.

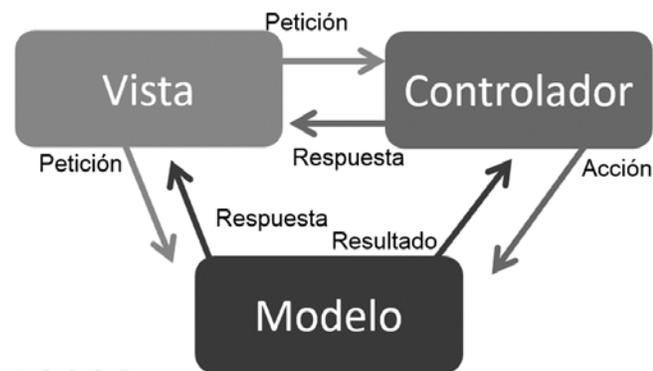


Figura 3. Vista del Patrón MVC.

2.5 Patrones de diseño de software utilizados para Fabulari.

En la ingeniería del software, un patrón constituye el apoyo para la solución a los problemas más comunes que se presentan durante las diferentes etapas del ciclo de vida del software. El uso de patrones de diseño permitirá describir fragmentos de diseño y reutilizar ideas de diseño. Los patrones de diseño comunican los estilos y soluciones consideradas como "buenas prácticas", que los expertos en el diseño orientado a objetos utilizan para la creación de sistemas (Larman, 2003).

2.5.1 Patrones para la asignación de responsabilidades (GRASP)

Los GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones (Larman, 2003). Para la implementación de la aplicación se hizo uso de los patrones siguientes:

El patrón **Controlador**, asigna a clases específicas la responsabilidad de controlar el flujo de eventos de la aplicación. Ello facilita la centralización de actividades (validaciones, seguridad, entre otros.). El controlador no las realiza, sino las delega en otras clases. Este patrón se ve reflejado en la clase "salaCP" donde es que se realiza el control de los objetos que son creados, editados o eliminados, que serán dirigidos a la base de datos, así como el uso correcto de mensajes de información dirigidos al usuario, en dicha clase se delega la responsabilidad del trabajo directo con la base de datos a la clase controladora del servidor "SalaCS".

Otro patrón utilizado es el **Creador (Creator)** que tiene la información necesaria para realizar la creación del objeto. Es decir este patrón ayuda a identificar quién debe ser el responsable de la creación (o instancia) de nuevos objetos o clases. Una de las consecuencias de usar este patrón es la visibilidad entre la clase creada y la clase creadora. Una ventaja es el bajo acoplamiento, lo cual supone facilidad de mantenimiento y reutilización. El mismo se ve aplicado en cualquier caso de agregación o composición de clases.

Para la asignación de responsabilidades en la aplicación se utilizó el patrón **Experto (Expert)** el cual indica que la clase que cuenta con la información necesaria para cumplir la responsabilidad es la responsable de manejar la información. De este modo se obtendrá un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento). Sus ventajas se centran en que mantiene el encapsulamiento, los objetos utilizan su propia información para llevar a cabo sus tareas, se distribuye el comportamiento entre las clases que contienen la información requerida y son más fáciles de entender y mantener.

Para fomentar la reutilización de código se tuvo en cuenta el patrón **Bajo Acoplamiento (Low Coupling)** que indica una menor dependencia entre clases. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases. También es utilizado el patrón **Alta Cohesión (High Cohesion)** debido a que los conceptos de cohesión y acoplamiento están íntimamente relacionados. Un mayor grado de cohesión implica un menor grado de acoplamiento. Maximizar el nivel de cohesión intramodular en todo el sistema resulta en una minimización del acoplamiento intermodular. Este patrón plantea que la información que almacena una clase debe de ser

coherente y debe estar (en la medida de lo posible) relacionada con la clase. Estos dos últimos patrones no deben tratarse de forma independiente pues poseen relaciones que mantienen el equilibrio entre clases, equilibra las responsabilidades y garantiza que las clases sean creadas con un buen diseño donde los objetos sean capaces de interactuar.

2.5.2 Patrones GoF

La línea base en el tema de patrones de diseño la impone el catálogo "*Design Patterns: Elements of Reusable Object-Oriented Software*", en este libro se presenta un conjunto de 23 patrones de diseño identificados a partir del estudio y la experiencia del grupo llamado Banda de los Cuatro o The Gang of Four (GOF): Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, quienes se dedicaron a analizar los problemas recurrentes en el desarrollo de software y realizaron una clasificación y agrupación a partir de dos criterios, su propósito y alcance, las categorías definidas son (Guerrero, y otros, 2013):

- Patrones Creacionales: Inicialización y configuración de objetos.
- Patrones Estructurales: Separan la interfaz de la implementación. Se ocupan de cómo las clases y objetos se agrupan, para formar estructuras más grandes.
- Patrones de Comportamiento: Más que describir objetos o clases, describen la comunicación entre ellos.

A partir de las categorías mencionadas se describen los patrones GOF empleados en el desarrollo del software.

2.5.2.1 Patrones de Comportamiento.

El patrón **Observador (Observer)** define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. En la aplicación se utiliza el Observador en la clase controladora "Transmisión", al utilizar el método "RefrescarListaSala()" que se ejecuta cuando un usuario entra a la sala de videoconferencia y es el encargado de mostrar a los demás usuarios la entrada del nuevo usuario a la sala de videoconferencia.

También se utilizó el patrón de comportamiento **Orden (Command)** que permite encapsular una petición de un objeto y luego realizar operaciones con el mismo. Dicho patrón se pone de manifiesto en la interfaz

“Insert_Sala.html” donde al utilizar, Angular.JS une la Vista con el Modelo a través de la directiva *data-ng-mode* la cual encapsula los datos necesarios para luego sean recogidos por la directiva *data-ng-submit = “create(salaForm.\$valid)”* para crear la sala (Ver Figura 4).

```
<section data-ng-controller="SalasController">
  <form name="salaForm" class="form-horizontal col-md-12" role="form" data-ng-submit="create(salaForm.$valid)" novalidate
    <div class="form-group col-md-5" ng-class="{ 'has-error' : submitted && salaForm.title.$invalid }">
      <label mean-token="create-title" class="col-md-3 control-label">Nombre</label>
      <div class="col-md-9">
        <input name="title" type="text" class="form-control" data-ng-model="title" id="title" placeholder="Nombre"
          <div ng-show="submitted && salaForm.title.$invalid" class="help-block">
            <p ng-show="salaForm.title.$error.required">El nombre es requerido</p>
            <p ng-show="salaForm.title.$error.minlength">La longitud mínima es 1</p>
            <p ng-show="salaForm.title.$error.maxlength">La longitud máxima es de 25 caracteres</p>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Figura 4. Representación del patrón Command.

2.5.2.2 Patrones Creacionales.

Durante el desarrollo de Fabulari se utilizó como patrón relacional **Instancia única (Singleton)** el cual garantiza que una clase sólo tenga una única instancia y proporciona un punto de acceso global a ella. Para inicializar la conexión a la base de datos de MongoDB (Ver Figura 5) en la clase controladora “SalaCP” para iniciar la conexión.

```
var mongoose = require('mongoose'),
    Sala = mongoose.model('Sala'),
    _ = require('lodash');

exports.sala = function(req, res, next, id) {
  Sala.load(id, function(err, sala) {
    if (err) return next(err);
    if (!sala) return next(new Error('No se pudo actualizar la sala ' + id));
    req.sala = sala;
    next();
  });
};
```

Figura 5. Representación del patrón de diseño Singleton.

2.6 Diagramas de secuencia

Cuando se tiene un esquema de las clases del diseño necesarias para realizar los CU, se debe describir cómo interactúan los correspondientes objetos del diseño. Esto se hace mediante diagramas de secuencia que contienen las instancias de los actores, los objetos del diseño y las transmisiones de mensajes entre estos, que participan en el CU. Si los CU tienen varios flujos, resulta útil crear un diagrama de secuencia para cada uno de ellos, para una mejor comprensión a la hora de realizarlos (Kruchten, 2003). A continuación se presentan los diagramas de secuencia de cada uno de los flujos del CU administrar salas de videoconferencia, los demás diagramas se podrán encontrar en el expediente de proyecto del producto STCV.

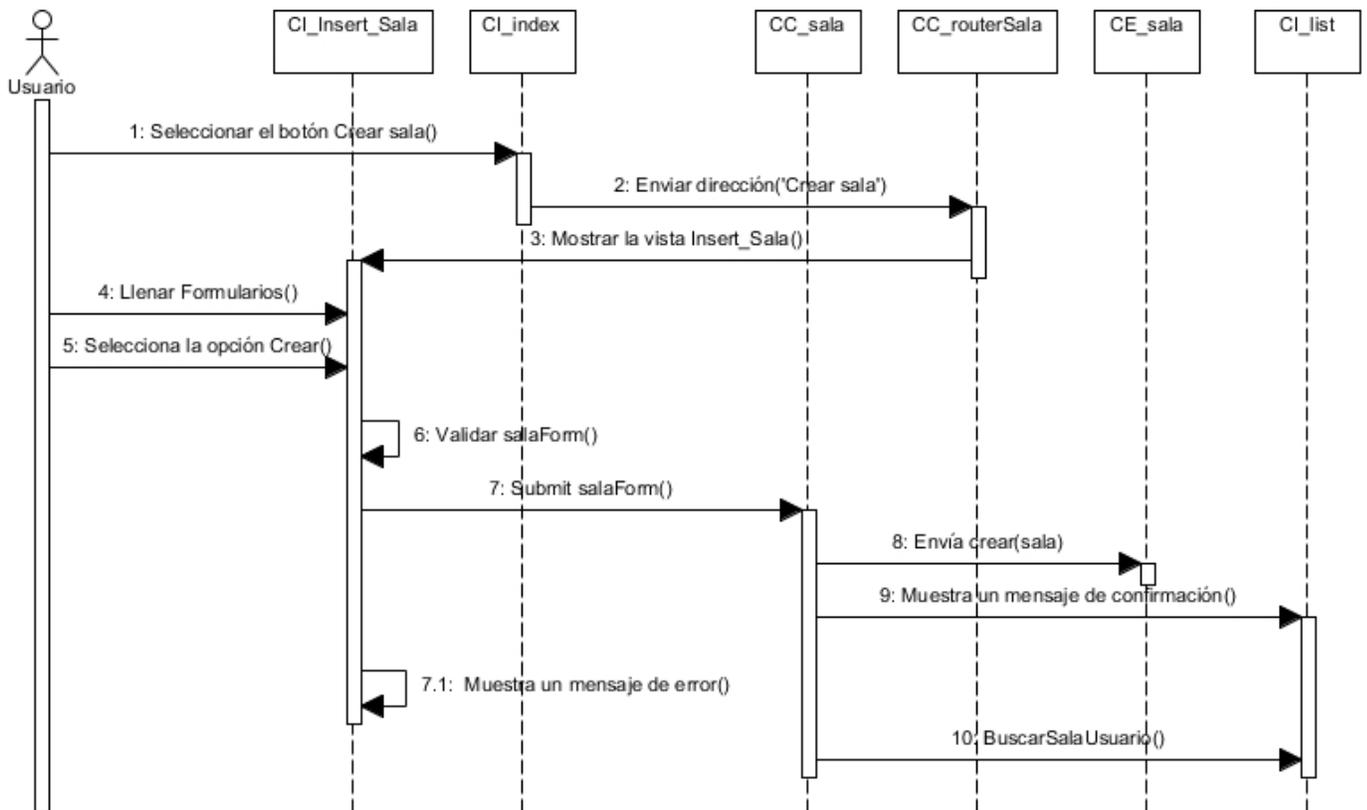


Figura 6. Diagrama de secuencia. Crear Sala de videoconferencia.

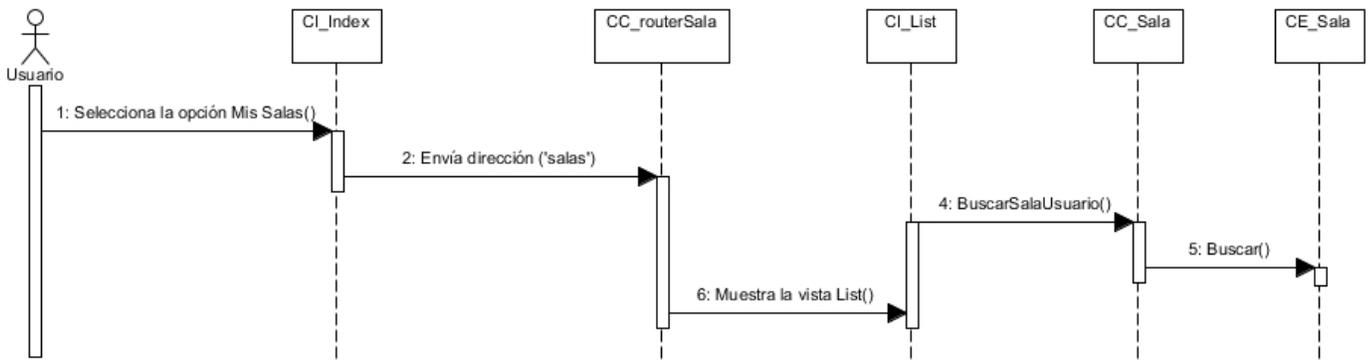


Figura 7. Diagrama de secuencia. Listar Sala de videoconferencia.

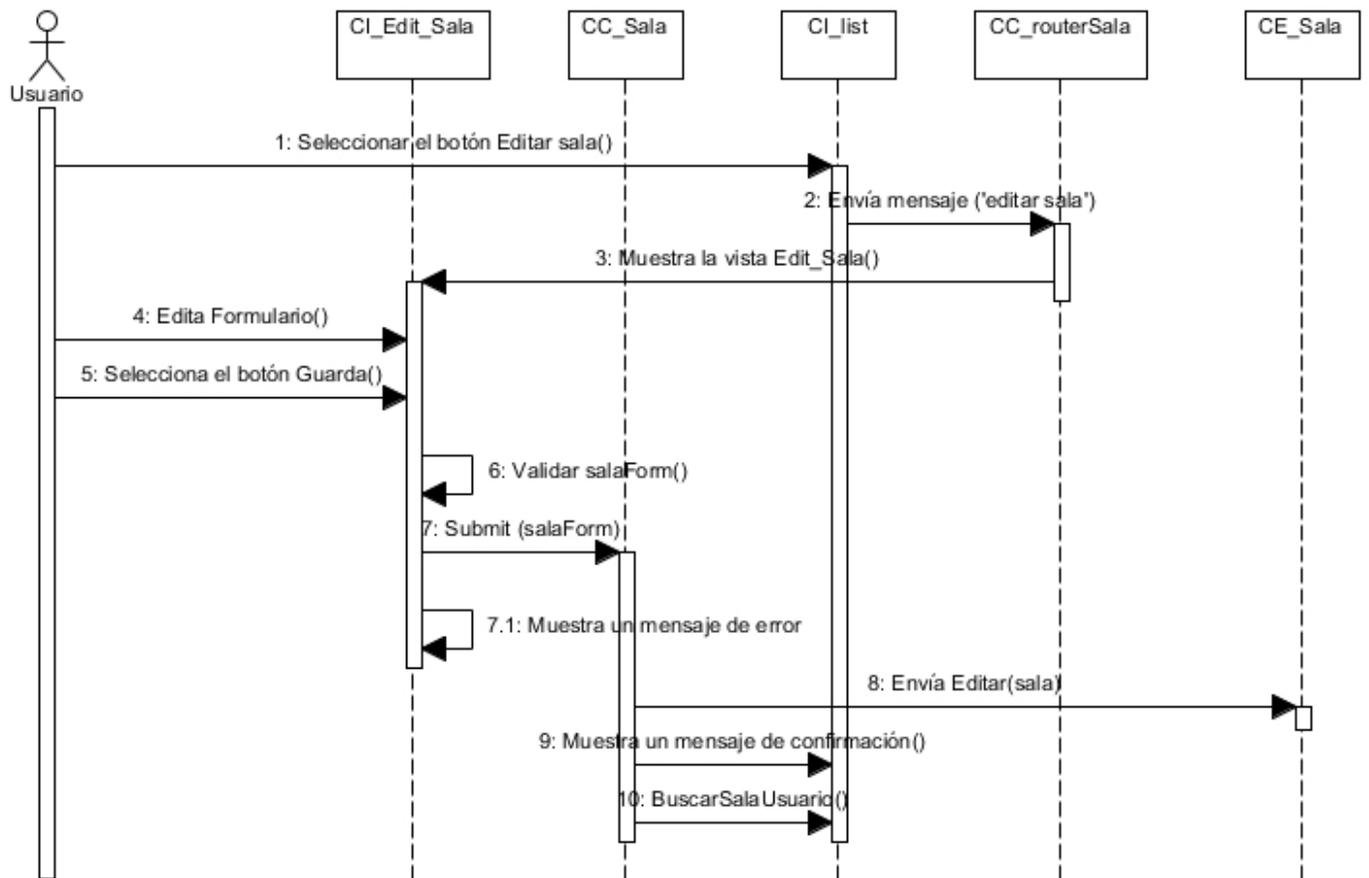


Figura 8. Diagrama de secuencia. Editar Sala de videoconferencia.

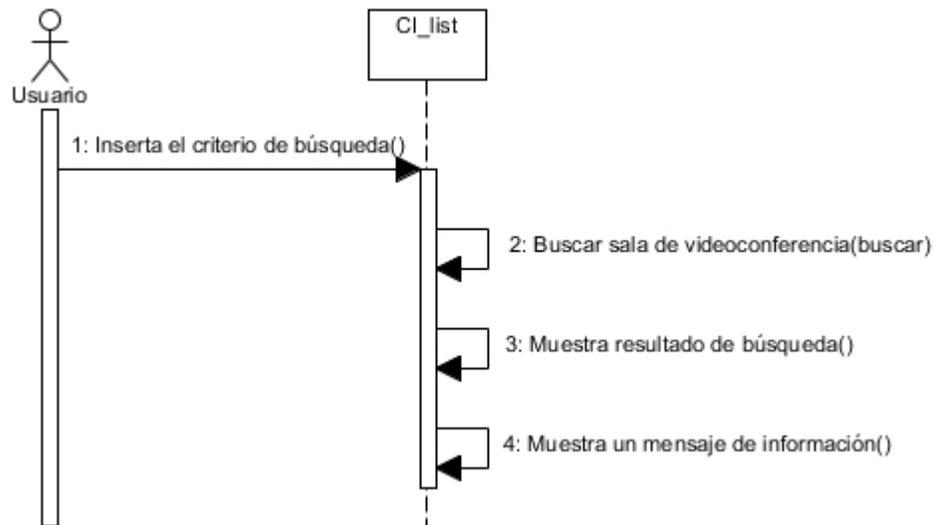


Figura 9. Diagrama de secuencia. Buscar Sala de videoconferencia.

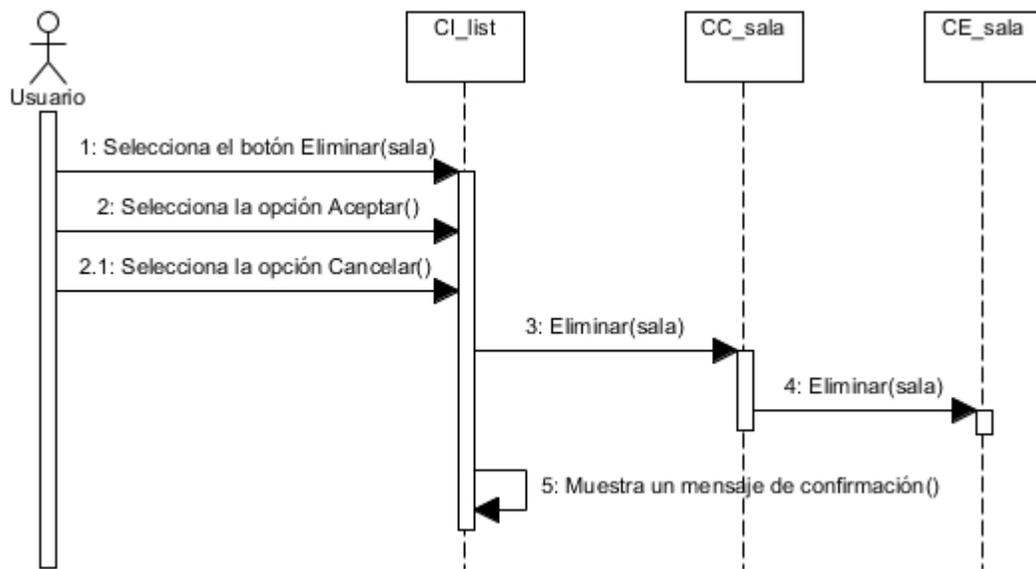


Figura 10. Diagrama de secuencia. Eliminar Sala de videoconferencia.

2.7 Diagrama de clases del diseño

Los diagramas de clases del diseño son un tipo de diagrama estático que describe gráficamente la estructura de una aplicación. Para contribuir a la calidad de *Fabulari* se hace uso de diagramas de clases del diseño, ya que sirven de guía a los desarrolladores al constituir una aproximación del sistema que se desea implementar (Pressman, 2010). A continuación se representa el diagrama de clases del diseño para la aplicación.

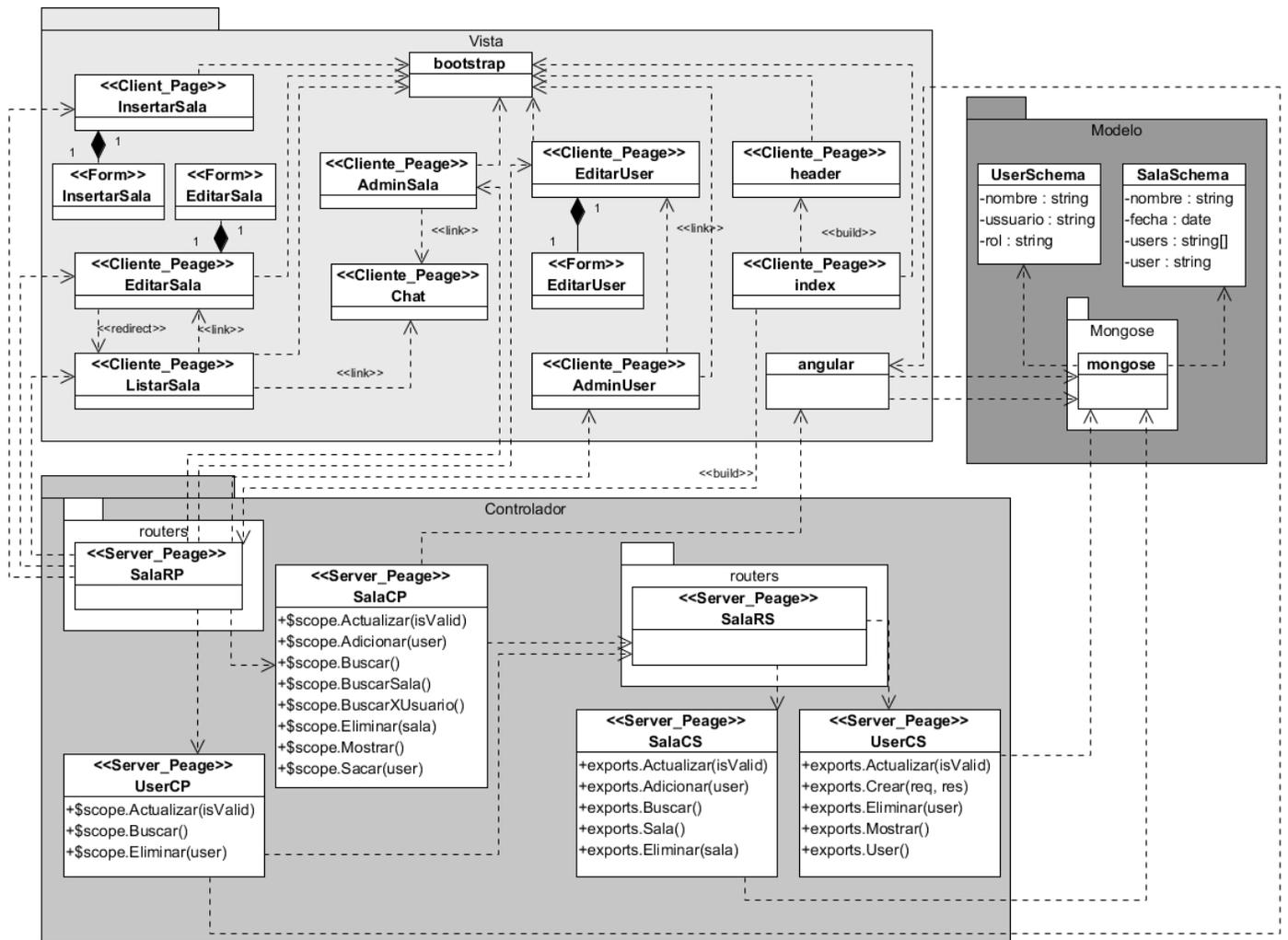


Figura 11. Diagrama de clases del diseño.

2.8 Conclusiones parciales.

La confección del diagrama de clases del dominio permite visualizar la relación existente entre los conceptos u objetos significativos en el dominio del problema. Con este artefacto, la obtención de los requisitos funcionales y no funcionales, de conjunto con el modelado del sistema, propician el entendimiento entre el equipo de desarrollo y el cliente de la aplicación en función de lo que el sistema debe realizar y las características que debe poseer. La descripción de los CU permite comprender el flujo de eventos que la aplicación debe cumplir para llevar a cabalidad los objetivos trazados. Se obtuvieron además los artefactos correspondientes al flujo de trabajo de Diseño los cuales sirven como guía para comenzar su implementación.

Capítulo 3. Implementación y Pruebas de *Fabulari*.

En el presente capítulo se tendrán en cuenta todos los aspectos del diseño del sistema con el fin de llevar a cabo el desarrollo de los flujos de trabajo de implementación y prueba. Se muestra la organización del sistema mediante el modelo de componentes el cual representa la vista estática del sistema y la situación física de los distintos componentes lógicos desarrollados a través del modelo de despliegue. Se describen los estilos de programación y los estándares de codificación empleados. Y por último se define el proceso de pruebas en el cual se verifica que todos los requisitos hayan sido implementados y funcionen correctamente.

3.1 Diagrama de Componente

Los diagramas de componentes ilustran las piezas del software que conformarán un sistema. Un diagrama de componentes tiene un nivel más alto de abstracción que un diagrama de clases, usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre dichos elementos. Es un grafo de componentes unidos a través de relaciones que pueden ser de compilación o de ejecución, y además se pueden representar las interfaces de esos componentes (SparxSystems, 2007). A continuación se muestra el diagrama de componentes de la aplicación (ver Figura 12):

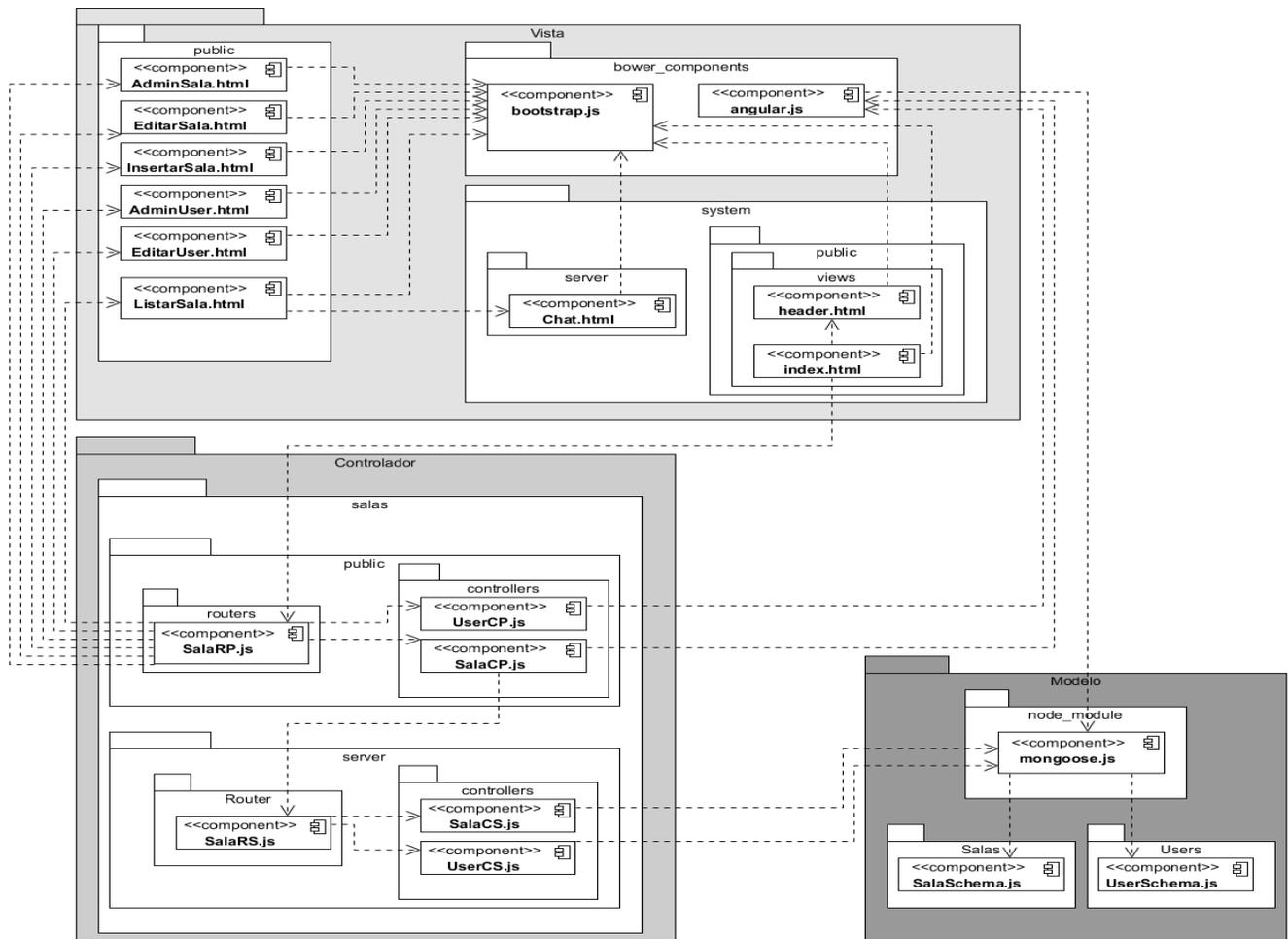


Figura 12. Diagrama de Componente.

3.2 Estándares de codificación

En la actualidad existen diferentes estándares de codificación cuya utilización favorece la comunicación fluida y directa entre los desarrolladores, permite la reutilización y el mantenimiento de los sistemas. Para implementar la solución propuesta se hacen uso de los siguientes estándares establecidos por los autores de esta investigación. Dichos estándares se describen a continuación:

Nombre de las funciones (métodos o funcionalidades de los controladores):

- Dentro de las especificaciones del framework está que cada una de los controladores deben finalizar con la palabra Controller. (ejemplo: SalaController)
- En el caso de los controladores de la vista, los nombres de acciones deben comenzar con \$scope. seguido por su nombre en notación lowerCamelCase. (ejemplo: \$scope.Sacar)

- En el caso de los controladores del servidor, los nombres de acciones deben comenzar con exports. seguido por su nombre en notación lowerCamelCase. (ejemplo: exports.Eliminar)
- Los nombres de las acciones deben especificar con la menor cantidad de palabras cuál es el objetivo de la acción, de ser posible estar en infinitivo.
- Antes de cada función se debe especificar su propósito a través de comentarios por vías que el desarrollador desee. Se recomienda usar: /* nueva línea * nueva línea */ (ver Figura 13).

```

/*Método para desmarcar un usuario a invitar*/
$scope.Sacar = function (user) {
    if ($scope.Marcas(user))
        $scope.sala.users.pop(user._id);
    else
        $scope.sala.users.push(user._id);
};

```

Figura 13. Representación de estándares de código utilizados.

Nombres de los modelos:

- Los nombres de los modelos deben estar expresados en notación UpperCamelCase y seguidos de la palabra Schema. (ejemplo: SalaSchema)
- No se deben utilizar guiones bajos en su nombre.
- Deben expresar con claridad cuál es el alcance y la responsabilidad de la clase.
- Los nombres de las clases no deben estar atados a las clases de las que se deriva, cada clase debe tener un significado por ella misma, no en dependencia de la clase de la que deriva.

Variables:

- Los nombres de las variables deben expresar claramente el contenido de la misma.
- Deben seguir la nomenclatura lowerCamelCase.
- Pueden estar referidas en singular o plural.
- Se definen al principio de las estructuras donde son utilizadas.
- En caso de que no se le asigne un valor inicial se deben inicializar con un valor que indique el tipo de dato más general al que debe pertenecer (ver Figura 14).

```

    /* SalaSchema modelo de Datos*/
    var SalaSchema = new Schema({
    } created: {
        type: Date,
        default: Date.now
    },
    },
    );

```

Figura 14. Representación de estándares de código utilizados.

3.3 Tratamiento de errores.

Un aspecto importante a tener en cuenta a la hora de desarrollar un software es el tratamiento de errores, tanto los generados por los usuarios como los producidos por el sistema. Las validaciones del negocio se encargan de controlar el flujo de los datos recibidos en el controlador para evitar la inconsistencia de la información almacenada en la base de datos.

Las validaciones realizadas en las vistas juegan un papel primordial en la entrada correcta de los datos a la aplicación. Tienen como función principal evitar que se introduzcan datos incorrectos en los diferentes campos de los formularios para impedir que datos inconsistentes o incorrectos lleguen al servidor. Con el objetivo de detectar y mostrar las alertas y mensajes de notificación de errores en la interfaz del sistema se utilizan las validaciones que provee AngularJS.

Por otro lado en cada acción implementada en MEAN se realiza un riguroso escrutinio de los valores de las variables que llegan a la funcionalidad en el servidor y en caso de detectarse errores son enviados en forma de cadena JSON⁵ a la interfaz para que sean notificados al usuario.

3.4 Modelo de despliegue

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador o un dispositivo (Marca, y otros, 2010).

⁵ Es un formato ligero para el intercambio de datos (Pérez, 2009).

El diagrama de despliegue propuesto para la aplicación web de videoconferencia *Fabulari* cuenta con tres nodos que responden a la distribución física de la aplicación (ver Figura 15). El nodo “Dispositivos Inteligentes” representa un equipo electrónico que puede ser un televisor, una tableta electrónica, un teléfono o una computadora, el cual permitirá la interacción con la aplicación, con el uso de otros dispositivos como bocinas, micrófonos y cámara de video, en caso de que lo requiera. El nodo “Dispositivos Inteligentes” establece una comunicación mediante el protocolo WebSocket con el nodo “Servidor Web”, el cual aloja la aplicación web de videoconferencia *Fabulari* que a su vez establece una comunicación con el nodo “Servidor base de datos”, donde se almacena toda la información gestionada por el sistema.

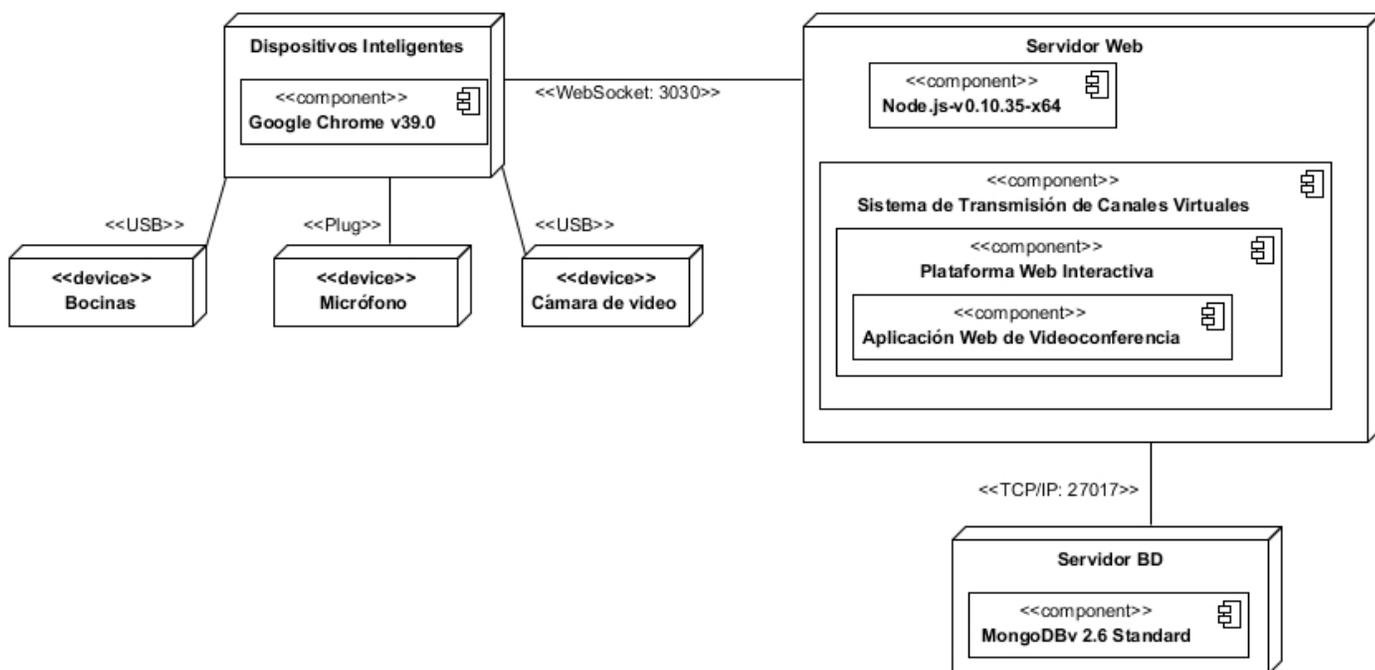


Figura 15. Diagrama de Despliegue.

3.5 Modelo de Pruebas

El modelo de pruebas describe principalmente cómo se prueban los componentes ejecutables en el modelo de implementación con pruebas de integración y de sistema, así como otros aspectos específicos del sistema (Kruchten, 2003). Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente (IEEE, 1995).

Una estrategia de prueba del software integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que dan como resultado una correcta construcción del software. La estrategia proporciona un mapa que describe los pasos que hay que llevar a cabo por parte de la prueba, cuándo se deben planificar y realizar esos pasos, y cuánto tiempo y recursos se van a requerir (Pressman, 2010).

Según (Pressman, 2010) la estrategia que se ha de seguir a la hora de evaluar la aplicación son una serie de niveles que se llevan a cabo secuencialmente. Al ensamblar o integrar los módulos para formar el paquete de software completo con el uso de la prueba de integración. Después se realizan las pruebas de validación proporcionando una seguridad final de que el software satisface todos los requisitos funcionales. Por último se realizan las pruebas del sistema, verificando que cada elemento encaja de forma adecuada y que alcanza la funcionalidad y el rendimiento del sistema total.

Para la aplicación web de videoconferencia *Fabulari* se aplicará la estrategia de prueba de los niveles de integración y de sistema. Las pruebas de integración tienen como objetivo construir una estructura de programa que esté de acuerdo a lo que dicta el diseño. Las pruebas del sistema están constituidas por una serie de pruebas, y aunque cada una tiene un propósito diferente, todas trabajan para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas.

3.5.1 Tipos de pruebas

Existen varios tipos de pruebas planteadas por (Pressman, 2010), algunas de ellas recomendadas específicamente para aplicaciones web, para la presente investigación por las características y funcionalidades de *Fabulari* se realizarán las que a continuación se explican:

Pruebas de integración ascendente: se aplican a medida que se va construyendo la arquitectura de la aplicación mientras que se llevan a cabo pruebas para descubrir errores asociados con la interfaz (Pressman, 2010). Para *Fabulari* se aplicará el enfoque incremental de integración ascendente.

Pruebas de función: ejercitan el contenido y las unidades funcionales dentro de la aplicación y se enfocan sobre un conjunto de pruebas que intentan descubrir errores en la misma. Para realizar las pruebas de función a *Fabulari* serán hará uso de la técnica de caja negra. Dicha técnica permite obtener un conjunto de condiciones de entrada que ejercitan completamente todos los requisitos funcionales de un programa. La prueba de caja negra intenta encontrar funciones incorrectas o ausentes, errores de interfaz, errores de estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y terminación. Para llevar a cabo esta técnica se realiza el Método de Partición de

Equivalencia que divide el campo de entrada de un programa en clases de datos. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduce así el número total de casos de prueba que hay que desarrollar. Un caso de prueba descubre de forma inmediata una clase de errores que de otro modo requerirían la ejecución de muchos casos antes de detectar el error genérico (Pressman, 2010).

Pruebas de desempeño o de rendimiento: se aplican para descubrir problemas debido a la falta de recursos en el lado del servidor, ancho de banda de red inapropiado, capacidades inadecuadas de bases de datos, defectuosas o débiles capacidades del sistema operativo, funcionalidades mal diseñadas y otros conflictos de hardware o software que pueden conducir a un pobre desempeño por parte del cliente o el servidor (Pressman, 2010). Como estrategia para la realización de las pruebas de rendimiento se hará uso de **la prueba de carga**, con el objetivo de examinar el comportamiento y la respuesta del servidor bajo varios niveles de carga, y **la prueba de esfuerzo** para determinar cuánta capacidad puede manejar el entorno de *Fabulari*.

3.6 Aplicación y resultado de las pruebas

La realización de las pruebas de caja negra a *Fabulari* requirió del diseño de los casos de prueba en correspondencia con las descripciones de casos de uso. A continuación se presenta la descripción del caso de prueba (DCP) correspondiente al caso de uso Gestionar sala de videoconferencia. El resto de los DCP correspondientes a los demás casos de uso se encuentran en los anexos de la presente investigación (Ver Anexo 4).

Tabla 5. Descripción de variables del Caso de Prueba del CU Administrar sala de videoconferencia.

No.	Nombre del Campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	Debe introducirse una cadena de caracteres de tipo string. El tamaño máximo requerido es de 25 caracteres.
2	Fecha	Date	No	Debe introducirse la fecha con el formato día, mes, año (dd-mm-aaaa).

3	Buscar	Campo de texto	Si	Debe introducirse una cadena de caracteres de tipo string.
4	Invitar	Checkbox	Si	Debe seleccionarse un campo.

Tabla 6. DCP para el CU: Administrar Sala de videoconferencia. Sección: Crear sala de videoconferencia.

Escenario	Descripción	Usuario	Fecha	Invitar	Buscar	Respuesta del sistema	Flujo central
EC 1.1 Crear nueva sala correctamente	Se intenta permitir al usuario crear una sala de videoconferencia correctamente.	V (Prueba)	V (11-Abril-2015)	V (Vacío)	V (Vacío)	Se adiciona una nueva sala al sistema. Se muestra un mensaje de confirmación "La operación ha sido realizada".	<ol style="list-style-type: none"> 1. Se selecciona desde el menú principal la opción "Crear nueva Sala". 2. Se muestra una vista con los campos a llenar de la sala de videoconferencia. 3. Se introducen los datos de la nueva sala de videoconferencia 4. Se presionar el botón "Crear". 5. Se verifican los datos.
EC 1.2 Crear nueva sala dejando campos vacíos.	Se intenta permitir al usuario crear una sala de videoconferencia dejando campos vacíos.	I (Vacío)	V (11-Abril-2015)	V (Vacío)	V (Vacío)	Muestra un mensaje de error "El nombre es requerido".	<ol style="list-style-type: none"> 1. Se selecciona desde el menú principal la opción "Crear nueva Sala". 2. Se muestra una vista con los campos a llenar de la sala de videoconferencia.

							<p>3. Se introducen los datos de la nueva sala de videoconferencia</p> <p>4. Se presionar el botón "Crear".</p> <p>5. Se verifican los datos.</p>
EC 1.3 Crear nueva sala introduciendo datos incorrectos.	Se intenta permitir al usuario crear una sala de videoconferencia introduciendo datos incorrectos.	I (Esta es una prueba de datos)	V (11-Abril-2015)	V (Vacío)	V (Vacío)	Muestra un mensaje de error "La longitud máxima es de 25 caracteres".	<p>1. Se selecciona desde el menú principal la opción "Crear nueva Sala".</p> <p>2. Se muestra una vista con los campos a llenar de la sala de videoconferencia.</p> <p>3. Se introducen los datos de la nueva sala de videoconferencia</p> <p>4. Se presionar el botón "Crear".</p> <p>5. Se verifican los datos.</p>

Utilizando los casos de prueba diseñados se realizaron dos iteraciones de pruebas para encontrar la mayor cantidad de errores en el funcionamiento de *Fabulari*.

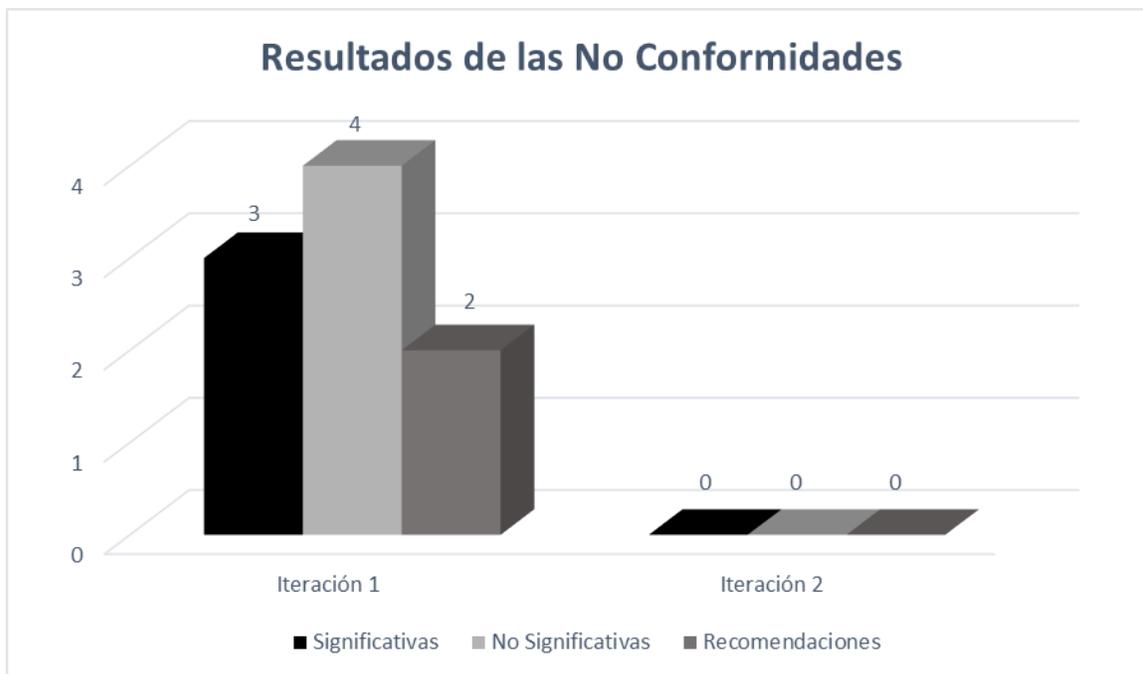


Figura 16. Gráfico de resultado de las no conformidades.

En el gráfico anterior (Figura 16) se evidencian nueve no conformidades detectadas al realizar la primera iteración, estas fueron clasificadas según su nivel de afectación en el funcionamiento del sistema, en significativas, no significativas y recomendaciones. Definidas como significativas aquellas no conformidades que afectaron el funcionamiento de la aplicación, las mismas fueron tres errores de funcionalidad. Como no significativas se detectaron cuatro errores de idioma y las restantes fueron recomendaciones o sugerencias en cambios a elementos del diseño. Al realizar la segunda iteración no se detectaron no conformidades (Ver anexo 5).

Para *Fabulari* se aplicó el enfoque incremental de integración ascendente ya que a medida que fueron desarrolladas las funcionalidades se agruparon y se coordinaron las entradas y salidas de los casos de prueba. Posteriormente se realizaron las pruebas al grupo de funcionalidades y al quedar probadas se continuó con el desarrollo de otras funcionalidades repitiendo el mismo enfoque. Una vez que se concluyó la implementación y se comprobó el correcto funcionamiento de *Fabulari* como aplicación, se realizaron las pruebas para constatar su integración a la Plataforma Web Interactiva. Donde se comprueba la autenticación de un usuario de la Plataforma Web Interactiva en *Fabulari* y su correcto funcionamiento dentro de la misma. A continuación se describen los resultados de las pruebas:

Se ejecutó la Plataforma Web Interactiva y se seleccionó entre las aplicaciones a *Fabulari* para ser ejecutada, instantáneamente *Fabulari* debería autenticar al usuario de forma automática, paso que no se

realizó pues existían sentencias mal declaradas en la configuración de la aplicación. Al concluir la corrección del error que dio fin a la primera iteración se prosiguió a comenzar una nueva iteración para comprobar si existían nuevos errores. Nuevamente se ejecutó la Plataforma Web Interactiva y se seleccionó a *Fabulari* para ser ejecutada, instantáneamente *Fabulari* logró autenticar automáticamente al usuario de la plataforma. Seguidamente se comprobaron las funcionalidades de *Fabulari* como la creación de sala y la utilización del chat, además se pusieron en marcha varias videoconferencias sin ocurrir ningún error. De esta manera culmina exitosamente la prueba comprobándose la correcta integración entre la aplicación web de videoconferencia *Fabulari* con la Plataforma Web Interactiva.

Pruebas de rendimiento: dichas prueba se realizarán haciendo uso de las estrategias de prueba de carga y prueba de esfuerzo con las que se realizará un análisis inicial a través de distintos escenarios. Las pruebas de rendimiento fueron realizadas con la herramienta Apache Jmeter versión 2.8 que brinda la posibilidad de poder realizar pruebas de rendimiento con varios hilos de peticiones HTTP a la aplicación. Las pruebas fueron realizadas con 5, 50 y 1000 muestras de datos, cuyas muestras serán el resultado de haber realizado 5, 50 y 1000 peticiones HTTP respectivamente a la aplicación, estas peticiones representan conexiones a la sala de videoconferencia. Como configuración para la realización de las pruebas se tomó como IP del servidor 127.0.0.1, el puerto 3030 y la ruta para realizar la prueba es la de una sala de videoconferencia como se muestra en la Figura 17.

The image shows the configuration interface for an HTTP request in Apache JMeter. The main configuration includes:

- Nombre:** Petición HTTP
- Comentarios:** (empty)
- Servidor Web:**
 - Nombre de Servidor o IP: 127.0.0.1
 - Puerto: 3030
- Petición HTTP:**
 - Implementación HTTP: (dropdown)
 - Protocolo: (dropdown)
 - Método: GET
 - Codificación del contenido: (dropdown)
 - Ruta: https://localhost:3030/chat/556e47353af4769c068b62fe
 - Options: Redirigir Automáticamente, Seguir Redirecciones, Utilizar KeepAlive, Usar 'multipart/form-data' para HTTP POST, Cabe
- Parameters:** Section for 'Enviar Parámetros Con la Petición' with columns for 'Nombre' and 'Valor', and buttons for 'Detail', 'Añadir', 'Add from Clipboard', 'Borrar', 'Up', and 'Down'.
- Files:** Section for 'Enviar un archivo Con la Petición' with a 'Nombre de Archivo:' field and buttons for 'Añadir', 'Navegar...', and 'Borrar'.
- Servidor Proxy:** Fields for 'Nombre de Servidor o IP:', 'Puerto:', and 'Nombre de U'.
- Tareas Opcionales:** (empty)

Figura 17. Representación de la configuración de las pruebas de resistencia.

Al realizar la prueba con 5 muestras se observa el comportamiento de *Fabulari*, donde se comprueba que tiene una rápida respuesta ante una carga de 5 usuarios. Se arrojó los siguientes resultados: una desviación (diferencia entre el valor máximo y mínimo del tiempo que demora la petición de un usuario) de 4ms, con una media de respuesta a las solicitudes de 1826 ms que se traduce en 1.826 segundos en el tiempo de respuesta (ver Figura 18).

Muestra #	Tiempo de comie...	Nombre del hilo	Etiqueta	Tiempo de Muest...	Estado	Bytes	Latency
1	20:59:25.048	Usuarios 1-5	Petición HTTP	1820		5717	1730
2	20:59:25.053	Usuarios 1-1	Petición HTTP	1828		5715	1723
3	20:59:25.055	Usuarios 1-4	Petición HTTP	1833		5713	1718
4	20:59:25.048	Usuarios 1-3	Petición HTTP	1827		5725	1733
5	20:59:25.048	Usuarios 1-2	Petición HTTP	1824		5721	1725

Scroll automatically?
 Child samples?
 No. de Muestras 5
 Última Muestra 1824
 Media 1826
 Desviación 4

Figura 18. Representación de resultados de las Pruebas de resistencia con 5 muestras.

Al realizar la prueba con 50 muestras se observa el comportamiento de *Fabulari*, donde se comprueba que muestra también una rápida respuesta ante la carga de 50 usuarios. Se arrojó los siguientes resultados: una media de respuesta a las solicitudes de 3525 ms, con una desviación de 861 ms, mostrando una estabilidad aceptable y no arrojando ningún error (ver Figura 19).

Muestra #	Tiempo de comie...	Nombre del hilo	Etiqueta	Tiempo de Mue...	Estado	Bytes	Latency
26	21:42:18.422	Pruebas 50 1-20	Petición HTTP	5255	▲	5713	4730
27	21:42:20.078	Pruebas 50 1-25	Petición HTTP	3645	▲	5717	3107
28	21:42:18.503	Pruebas 50 1-19	Petición HTTP	5222	▲	5717	4689
29	21:42:20.810	Pruebas 50 1-27	Petición HTTP	2988	▲	5717	2398
30	21:42:20.571	Pruebas 50 1-29	Petición HTTP	3262	▲	5713	2763
31	21:42:20.612	Pruebas 50 1-42	Petición HTTP	3227	▲	5715	2834
32	21:42:20.126	Pruebas 50 1-39	Petición HTTP	3711	▲	5721	3303
33	21:42:20.587	Pruebas 50 1-35	Petición HTTP	3386	▲	5711	3091
34	21:42:20.831	Pruebas 50 1-32	Petición HTTP	3152	▲	5715	2791
35	21:42:20.863	Pruebas 50 1-44	Petición HTTP	3122	▲	5713	2825
36	21:42:19.940	Pruebas 50 1-36	Petición HTTP	4090	▲	5713	3721
37	21:42:20.817	Pruebas 50 1-49	Petición HTTP	3233	▲	5719	2838
38	21:42:20.753	Pruebas 50 1-46	Petición HTTP	3336	▲	5715	2898
39	21:42:20.629	Pruebas 50 1-33	Petición HTTP	3364	▲	5717	3074
40	21:42:19.953	Pruebas 50 1-26	Petición HTTP	4144	▲	5721	3699
41	21:42:20.138	Pruebas 50 1-34	Petición HTTP	3967	▲	5713	3851
42	21:42:20.216	Pruebas 50 1-45	Petición HTTP	3894	▲	5719	3474
43	21:42:20.626	Pruebas 50 1-47	Petición HTTP	3509	▲	5713	3008
44	21:42:20.375	Pruebas 50 1-31	Petición HTTP	3788	▲	5721	3315
45	21:42:20.833	Pruebas 50 1-41	Petición HTTP	3329	▲	5713	3193
46	21:42:20.775	Pruebas 50 1-48	Petición HTTP	3401	▲	5717	3246
47	21:42:18.735	Pruebas 50 1-18	Petición HTTP	5458	▲	5713	5281
48	21:42:20.261	Pruebas 50 1-40	Petición HTTP	3925	▲	5715	3392
49	21:42:20.921	Pruebas 50 1-50	Petición HTTP	3315	▲	5715	3092
50	21:42:20.461	Pruebas 50 1-43	Petición HTTP	3777	▲	5719	3548

Scroll automatically?
 Child samples?
 No. de Muestras 50
 Última Muestra 3777
 Media 3525
 Desviación 861

Figura 19. Representación de resultados de las Pruebas de resistencia con 50 muestras.

Por último se ejecutó la prueba con 1000 muestras como el caso más crítico con el objetivo de verificar el esfuerzo de *Fabulari* arrojando como resultado una media de 2673 ms, con una desviación de 1737ms, sin mostrar ningún error o colapso (ver Figura 20 y Figura 21).

Muestra #	Tiempo de comie...	Nombre del hilo	Etiqueta	Tiempo de Mue...	Estado	Bytes	Latency
976	22:01:44.653	Grupo de Hilos 1...	Petición HTTP	3038	▲	5719	2960
977	22:01:44.875	Grupo de Hilos 1...	Petición HTTP	2819	▲	5717	2749
978	22:01:44.915	Grupo de Hilos 1...	Petición HTTP	2789	▲	5717	2703
979	22:01:44.499	Grupo de Hilos 1...	Petición HTTP	3206	▲	5717	3121
980	22:01:44.867	Grupo de Hilos 1...	Petición HTTP	2855	▲	5711	2745
981	22:01:43.801	Grupo de Hilos 1...	Petición HTTP	3925	▲	5717	3833
982	22:01:43.725	Grupo de Hilos 1...	Petición HTTP	4030	▲	5713	3941
983	22:01:44.419	Grupo de Hilos 1...	Petición HTTP	3340	▲	5713	3239
984	22:01:45.016	Grupo de Hilos 1...	Petición HTTP	2758	▲	5715	2654
985	22:01:44.991	Grupo de Hilos 1...	Petición HTTP	2810	▲	5715	2732
986	22:01:44.690	Grupo de Hilos 1...	Petición HTTP	3129	▲	5717	3068
987	22:01:44.851	Grupo de Hilos 1...	Petición HTTP	3023	▲	5717	2951
988	22:01:44.845	Grupo de Hilos 1...	Petición HTTP	3050	▲	5719	2973
989	22:01:44.461	Grupo de Hilos 1...	Petición HTTP	3465	▲	5713	3365
990	22:01:44.472	Grupo de Hilos 1...	Petición HTTP	3473	▲	5713	3375
991	22:01:44.519	Grupo de Hilos 1...	Petición HTTP	3445	▲	5715	3343
992	22:01:45.012	Grupo de Hilos 1...	Petición HTTP	2978	▲	5713	2886
993	22:01:44.836	Grupo de Hilos 1...	Petición HTTP	3051	▲	5713	2968
994	22:01:43.715	Grupo de Hilos 1...	Petición HTTP	4183	▲	5711	4079
995	22:01:44.550	Grupo de Hilos 1...	Petición HTTP	3393	▲	5717	3280
996	22:01:44.446	Grupo de Hilos 1...	Petición HTTP	3506	▲	5721	3400
997	22:01:44.782	Grupo de Hilos 1...	Petición HTTP	3212	▲	5715	3068
998	22:01:44.633	Grupo de Hilos 1...	Petición HTTP	3377	▲	5721	3314
999	22:01:45.341	Grupo de Hilos 1...	Petición HTTP	2693	▲	5717	2643
1000	22:01:43.679	Grupo de Hilos 1...	Petición HTTP	4342	▲	5717	3431

Scroll automatically?
 Child samples?
 No. de Muestras 1000
 Última Muestra 4342
 Media 2673
 Desviación 1737

Figura 20. Representación de resultados de las Pruebas de resistencia con 1000 muestras.

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
Peticion HTTP	1000	3601	3345	6256	221	11098	0,00%	24,4/sec	136,3
Total	1000	3601	3345	6256	221	11098	0,00%	24,4/sec	136,3

Figura 21. Representación de resultados de las Pruebas de resistencia. Resultados generales con 1000 muestras.

Con los resultados arrojados se demuestra que la aplicación en un ámbito de trabajo normal no colapsará, ya que se estima que las peticiones no superarán las 1000 muestras en un mismo instante de tiempo.

3.7 Conclusiones parciales

Mediante el diagrama de componentes se representó una vista estática de la aplicación, mostrando la organización y dependencia que existe entre los componentes físicos que se necesitan para ejecutar la misma. La realización del diagrama de despliegue permitió comprender la distribución física del sistema. El uso de los estándares de codificación y estilos de programación permiten el entendimiento del código por otros programadores que no sean del equipo de desarrollo en caso de mejoras a la aplicación en un futuro. Con el desarrollo del proceso de pruebas se pudo identificar y resolver los errores en la implementación aumentando la calidad del producto final obtenido. También se comprobó que el mismo satisface las funcionalidades requeridas, cumple con los requisitos no funcionales planteados y se integra correctamente con la Plataforma Web Interactiva.

Conclusiones Generales

Una vez culminada la investigación se puede afirmar que se le dio cumplimiento a los objetivos planteados, arribando a las siguientes conclusiones:

- El estudio de los principales conceptos relacionados a las videoconferencias y su transmisión permitió sentar las bases para el desarrollo de la aplicación web interactiva *Fabulari*.
- El estado del arte referido a la transmisión de videoconferencias sobre la web permite afirmar que las soluciones que existen, que de alguna manera tributan a la investigación, no resuelven la problemática planteada, evidenciándose la necesidad de esta investigación.
- Las herramientas, tecnologías y lenguajes propuestos para la construcción de la aplicación, se corresponden con las políticas de soberanía tecnológica que impulsa la universidad y el país.
- Los artefactos generados durante el proceso de desarrollo del software permitirán un mejor entendimiento de *Fabulari* en caso de mejoras en el futuro.
- Los 17 Requisitos Funcionales que se definieron fueron debidamente implementados y se incluyó en la propuesta de solución las exigencias de todos los Requisitos No Funcionales detectados.
- La utilización de patrones y estándares de codificación promueve buenas prácticas en el desarrollo de la solución siendo más entendible y proporcionando uniformidad en la implementación.
- Los diseños de Casos de Prueba desarrollados, como parte de las pruebas de Caja Negra y la ejecución de las pruebas a nivel de integración y de sistema permitieron validar los requisitos de la aplicación con las funcionalidades implementadas y su correcta integración con la Plataforma Web Interactiva.
- Como resultado se obtuvo la aplicación web de videoconferencia *Fabulari*, la cual garantiza la comunicación entre usuarios de la Plataforma Web Interactiva haciendo uso del protocolo WebSocket.
- La utilización de *Fabulari* le permitirá a los usuarios de la Plataforma Web Interactiva realizar reuniones e intercambios de ideas, experiencias y conocimientos para solucionar problemas en conjunto.

Recomendaciones

Se recomienda incorporar a *Fabulari* una funcionalidad de *Closed Caption* o Subtitulado Oculito para extender su uso a personas con discapacidades auditivas.

Referencias Bibliográficas

- Acevedo, Edwin Jovanny Clavijo, Hernández Chacón, Sorey y Cardoza Vásquez, Edison. 2015.** *Tecnología Streaming para Radio Digital Universitaria*. Bucaramanga-Colombia : Revista ESAICA, 2015. Vol. 1.
- Acosta , María del Rocío Reyes, y otros. 2015.** *Uso de las tecnologías en telecomunicación para la videoconferencia dual con aplicación en la medicina*. México : s.n., 2015. Vol. 8.
- Alvarez, Carlos de Zayas. 1995.** *Metodología de la Investigación Científica*. Santiago de Cuba : s.n., 1995.
- Alvarez, Migue Angel. 2012.** *Manual de jQuery*. 2012.
- Amazon. 2015.** AngularJS: Novice to Ninja: Sandeep Panda: 9780992279455: Amazon.com: Books. [En línea] 2015. <http://www.amazon.com/AngularJS-Novice-Ninja-Sandeep-Panda/dp/0992279453>.
- Baset, Salam A. and Henning Schulzrinn. 2004.** *An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol*. New York : s.n., 2004.
- Cerviño, J., y otros. 2008.** *Marte 3.0: Una videoconferencia 2.0*. 2008.
- Clements, P, Bass, L y Kazman, R. 2003.** *Software Architecture in Practice*. . s.l. : SEI Series in Software Engineering: Addison Wesley, 2003.
- Díaz, Genaro Solís, y otros. 2000.** *ESTUDIO DEL DISEÑO DE SISTEMA DE VIDEOCONFERENCIA PARA INTERNET Y LA RED DIGITAL DE SERVICIOS INTEGRADOS*. 2000.
- Express. 2015.** Express - Node.js web application framework. [En línea] 2015. <http://expressjs.com/>.
- González, Zenona Aponcio. 2010.** Videoconferencia - Presentation Transcript. [En línea] 2010. [Citado el: 10 de 11 de 2014.] <http://www.slideshare.net/ansaca/videoconferencia>.
- Guerrero, Carlos A. , Suárez, Johanna M. y Gutiérrez, Luz E. . 2013.** Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. [En línea] 2013. [Citado el: 28 de 3 de 2014.] http://www.scielo.cl/scielo.php?pid=S0718-07642013000300012&script=sci_arttext.0718-0764.
- Halsall, Fred. 2005.** *Computer Networking and the Internet*. Fifth edition. 2005.

HTML5. 2015. *CSS3 HTML5.* [En línea] 2015. [Citado el: 10 de 01 de 2015.] <http://html5.dwebapps.com/que-es-css3/>.

IEEE. 1995. *IEEE Recommended Practice for the Adoption of Computer-Aided Software Engineering (CASE) Tools.* 1995.

INEI. 1999. *Herramientas Case: COLECCION CULTURA INFORMATICA.* s.l. : Talleres de la Oficina de Impresiones de la Oficina Técnica de Difusión del Instituto Nacional de Estadística e Informática (INEI), 1999.

JetBrains. 2015. *PhpStorm.* [En línea] 2015. [Citado el: 10 de enero de 2015.] <https://www.jetbrains.com/phpstorm/whatsnew/>.

jWebSockets. 2014. *jWebSockets.* [En línea] 2014. [Citado el: 20 de Mayo de 2014.] <https://jwebsocket.org/>.

Kruchten, Philippe. 2003. *Rational Unified Process, The: An Introduction, Third Edition.* s.l. : Addison Wesley, 2003. ISBN: 0-321-19770-4.

Larman, Craig. 2003. *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado.* 2da Edición. 2003.

Larousse. 2015. *Diccionario Manual de la Lengua Española.* s.l. : Larousse, 2015.

Libreeducación. 2013. *¿Qué es una Sala Virtual? y sus usos habituales - Libreeducacion.* [En línea] 2013. [Citado el: 18 de diciembre de 2014.] <http://libreeducacion.com/sala-virtual/>.

López, Alberto Tallón. 2011. *Microlopez.* 2011.

López, José Alejandro Pérez. 2010. *Internet's Information as a Video Signal and its Editor.* Bogotá : s.n., 2010. ISSN 0120-4823.

Manolakis, Mg. Laura y Ceballos, Lic. Marcela . 2015. *ulas presenciales y aulas virtuales: Espacios que forman.* Quilmes : s.n., 2015.

Marca, Hugo y Quisbert, Nancy. 2010. *Diagrama de Despliegue.* [En línea] 2010. [Citado el: 14 de 03 de 2014.] virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc.

MEAN.IO. 2015. *MEAN.IO - Full-Stack JavaScript Using MongoDB, Express, AngularJS, and Node.js.* [En línea] 15 de febrero de 2015. <http://www.meanio.org/>.

- Millán, Ramón Jesús Tejedo. 2014.** *¿Qué es WebRTC (Web Real-Time Communication)?* 2014.
- MongoDB. 2015.** MongoDB. [En línea] 2015. <https://www.mongodb.org/>.
- Mozilla Foundation. 2015.** developer.mozilla.org. [En línea] 2015. [Citado el: 25 de enero de 2015.] <https://developer.mozilla.org/es/docs/Web/JavaScript>.
- Muñoz, Arturo de la Torre Monzón. 2013.** *Introducción a Node.JS a través de Koans*. Madrid : s.n., 2013.
- Node Hispano . 2014.** Node Hispano. [En línea] 2014. [Citado el: 12 de Enero de 2014.] <http://www.nodehispano.com/2012/10/linkedin-migra-desde-rails-hacia-node-nodejs/>.
- NodeJS. 2014.** Node.js. [En línea] 2014. [Citado el: 1 de 12 de 2014.] <http://www.nodejs.org/>.
- Nuñez, Jenny Alexandra Villacis. 2007.** *DESARROLLO DE UNA VIDEOCONFERENCIA REMOTA MULTIPLATAFORMA UTILIZANDO EL PROTOCOLO DE INCIO DE SESION*. Latacunga : s.n., 2007.
- Paradigm, Visual. 2014.** Visual Paradigm. *Visual Paradigm*. [En línea] 2014. <http://www.visual-paradigm.com/>.
- Pardo, Hugo Kuklinski. 2006.** *Un modelo de aplicación web institucional universitaria*. Barcelona : s.n., 2006.
- Pressman, Roger S. 2010.** *Ingeniería de Software. Un enfoque Práctico*. 7ma Edición. 2010.
- Recol. 2007.** RECOL. [En línea] 2007. http://www.recol.es/index.php?option=com_content&task=view&id=110&Itemid=417.
- Rena. 2015.** *RENa - Cuarta etapa - Informática - Lenguajes de programación*. [En línea] 2015. <http://www.rena.edu.ve/cuartaEtapa/Informatica/Tema13.html>.
- Ribas, Miquel Oliver. 2012.** *LA VIDEOCONFERENCIA EN EL CAMPO EDUCATIVO. TÉCNICAS Y PROCEDIMIENTOS*. s.l. : Universidad de las Islas Baleares, 2012.
- Riehle, Dirk. 2000.** *Framework Design. A Role Modeling Approach*. [En línea] 2000. <http://www.riehle.org/computer-science/research/dissertation/diss-a4.pdf>.
- Ruiz, Miguel Ángel Orbegoso. 2011.** *¿Qué es una conferencia?* 2011.

Skvorc, Bruno. 2015. Best PHP IDE in 2014 – Survey Results. [En línea] 2015. [Citado el: 13 de febrero de 2015.] <http://www.sitepoint.com/best-php-ide-2014-survey-results/>.

SparxSystems. 2007. Diagrama de Componentes UML 2. [En línea] Sparx Systems Pty Ltd., 2007. [Citado el: 11 de 03 de 2014.] http://www.sparxsystems.com.ar/resources/tutorial/uml2_componentdiagram.html.

Tejedor, Ramón Jesús Millán. 2014. *WebRTC (Web Real-Time Communications)*. 2014.

UCOL. 2006. Las herramientas CASE. [En línea] Universidad de Colima, 2006. [Citado el: 12 de 03 de 2014.] http://docente.ucol.mx/al961223/public_html/centro6.htm.

Vega, John Freddy y Van Der Henst, Christian. 2011. *Guía HTML5, el presente de la web*. 2011.

Visual Paradigm. 2014. Visual Paradigm. What VP-UML Provides. [En línea] 2014. [Citado el: 25 de Febrero de 2014.] <http://www.visual-paradigm.com/product/vpuml/provides>.

Voztele. 2014. Funcionalidades de OIGAA Meeting, la videoconferencia profesional. [En línea] 6 de 11 de 2014. <http://www.voztele.com/es/videoconferencia-videollamada-oigaa-meeting-funcionalidades>.

WebSocket.2014. *WebSocket.org -- A WebSocket Community*. [En línea] 2014. [Citado el: 4 de 12 de 2014.] <https://www.websocket.org/>.