

Universidad de las Ciencias Informáticas

Facultad 6



**Título: Componente audiovisual para el módulo
Teleconsulta Genética del sistema alasMEDIGEN.**

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

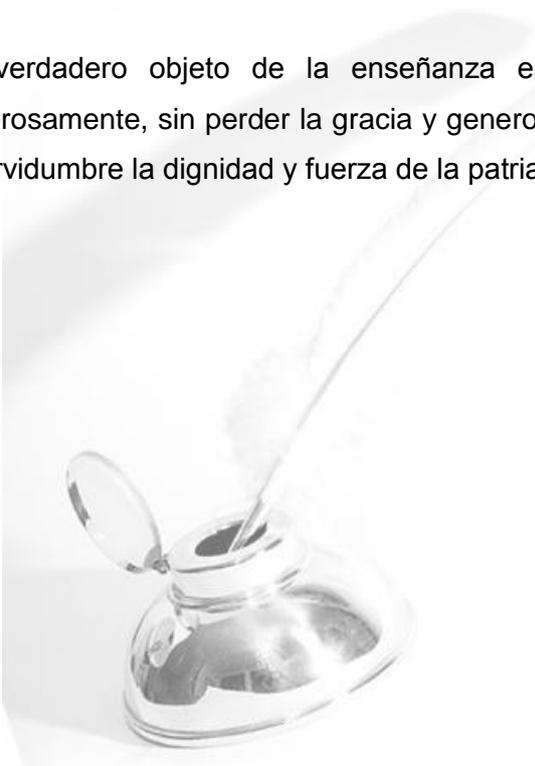
Autora: Sadys María Pérez Calleja

Tutores: Dayana Joseph Smarth

Yudiel La Rosa González

“El verdadero objeto de la enseñanza es preparar al hombre para que pueda vivir por sí decorosamente, sin perder la gracia y generosidad del espíritu, y sin poner en peligro con su egoísmo o servidumbre la dignidad y fuerza de la patria.”

José Martí.



DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Sadys María Pérez Calleja

Ing. Dayana Joseph Smart

Ing. Yudiel La Rosa González



Tutor:

Ing. Dayana Joseph Smarth.

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: djoseph@uci.cu

Tutor:

Ing. Yudiel La Rosa González.

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: ylarosag@uci.cu

Consultante:

Ing. Duany Baró Menéndez.

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: dbaro@uci.cu



AGRADECIMIENTOS

A la Revolución y a Fidel por darme la oportunidad de cursar esta carrera.

A las personas que más quiero en este mundo, a mis padres, por ser la luz que siempre guió mis pasos, por su dedicación y apoyo constante.

A Ricardo por su extremada paciencia, por entenderme y apoyarme siempre.

A Dayana y a Yudiel por ser mis tutores, por ayudarme con sus críticas y consejos, por soportar mis quejas y escucharme siempre, por su apoyo incansable en todo momento.

A Dayana porque me ha soportado todo este tiempo, por su amistad.

A Duany por su enorme ayuda, por dedicarme su tiempo y por su especial atención.

A Leovan y Rubizel, por su ayuda y apoyo en todo momento.

A Eddy, Jessie, Lisday, Luis Miguel y Eyllin por ser mis amigos en estos cinco años.

A todos los amigos de la familia que han ayudado a mis padres, cuando yo no he estado.

A todas las amistades de la universidad, con las que he podido compartir momentos únicos.

A todas las personas que han contribuido de alguna manera a mi formación profesional a través de sus enseñanzas y lecciones, a todas gracias por guiarme siempre



DEDICATORIA

A mis padres:

Por ser las personas que más me aman, las más lindas, por ser los mejores padres que la vida me haya podido dar jamás, por su amor incondicional y su cariño desmedido. Por haberme inculcado los principios, los valores, la perseverancia y el empeño que caracterizan la persona que soy. Por ser el mejor regalo que me ha dado la vida, mi mayor alegría, mis ojos, mi gran virtud, por ser la mayor muestra de amor y bondad que dios ha guardado para mí. Ustedes son lo que más amo.

A mis abuelos, en especial a mi abuelo Gerardo para que desde el cielo vea que siguen existiendo Callejas para rato.



RESUMEN

El potencial de las Tecnologías de la Informática y las Comunicaciones (TIC) en el sector de la salud, permite optimizar la organización y la calidad de la atención a los pacientes. La Telemedicina es una de las herramientas que se desarrolló con este objetivo. Esta está dividida en servicios de tratamiento, de diagnóstico y de información o educación. Entre los servicios de diagnóstico se encuentra la Teleconsulta, que permite la asistencia médica a distancia, actividad que se desarrolla a nivel mundial.

La solución informática alasMEDIGEN: “Sistema Informático de Genética Médica” contiene un conjunto de módulos que permiten la integración y gestión de la información de distintos estudios genéticos que se realizan en el país. Uno de estos módulos es Teleconsulta Genética, que posibilita la realización de conferencias basadas en texto, pero los especialistas no pueden tener un contacto visual y acústico con el paciente, incidiendo en la atención médica que estos últimos requieren.

La presente investigación tiene como objetivo desarrollar un componente audiovisual para el módulo Teleconsulta Genética del sistema alasMEDIGEN, que permitirá mejorar la comunicación entre los genetistas, desarrollando las Teleconsultas mediante videoconferencias. Este componente brindará una interfaz amigable para realizar las reuniones virtuales, permitirá guardar las videoconferencias y sus datos, para que puedan ser consultados posteriormente. Ofrecerá la posibilidad a los especialistas de hacer los diagnósticos de manera más precisa y generalizar el conocimiento entre ellos.

Palabras claves: Teleconsulta, genética.



ÍNDICE

AGRADECIMIENTOS.....	IV
DEDICATORIA	V
RESUMEN.....	VI
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	4
1.1 Teleconsultas.....	4
1.2 Situación actual del módulo Teleconsulta Genética de alasMEDIGEN.....	6
1.3 Antecedentes.....	7
1.3.1 Sistemas de Teleconsulta en el ámbito internacional.....	7
1.3.2 Sistemas de Teleconsulta en el ámbito nacional.....	9
1.4 Herramientas, tendencias y tecnologías actuales.....	10
1.4.1 Tecnología Media Streaming.....	10
1.4.2 Estándares de codificación para la transmisión de video y audio.....	12
1.4.3 Metodología de desarrollo.....	13
1.4.4 Lenguaje de Modelado.....	14
1.4.5 Lenguajes de programación utilizados.....	14
1.4.6 Frameworks de desarrollo.....	15
1.4.7 Herramienta CASE.....	16
1.4.8 IDE utilizados.....	16
1.4.9 Sistema Gestor de Base de Datos (SGBD).....	16
1.4.10 Servidor de aplicaciones Web.....	17
1.4.11 Patrones arquitectónicos utilizados.....	17
1.5 Conclusiones.....	19

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.	20
2.1 Modelado del negocio.	20
2.1.1 Descripción del modelo del negocio.	20
2.1.2 Actores y trabajadores del negocio.....	20
2.1.3 Caso de Uso del negocio.	21
2.1.4 Descripción textual del Caso de Uso del negocio.	21
2.1.5 Diagrama de actividades del negocio.	23
2.1.6 Modelo de Objetos.	24
2.2 Propuesta del sistema.....	24
2.3 Levantamiento de requisitos.....	24
2.3.1 Requisitos funcionales.....	24
2.3.2 Requisitos no funcionales:.....	25
2.4 Diagrama de Casos de Uso del sistema.....	27
2.4.1 Descripción de los actores del sistema.....	28
2.4.2 Descripción de los Casos de Uso del sistema.	28
2.5 Conclusiones.	35
CAPÍTULO 3: DISEÑO DEL SISTEMA.	37
3.1 Arquitectura.....	37
3.2 Vista Lógica de la arquitectura.	37
3.3 Diagramas de Clases del Diseño.	39
3.4 Modelo de datos.....	41
3.5 Modelo de clases persistentes.	42
3.6 Diagramas de secuencias.	44
3.7 Patrones de diseño.	46
3.7.1 Patrones de asignación de responsabilidades.....	46
3.7.2 Patrones GOF	47

3.8 Diagrama de Despliegue.....	47
3.9 Conclusiones.....	48
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.....	49
4.1 Diagramas de Componentes.....	49
4.2 Estándares de código.....	50
4.3 Pruebas.....	51
4.4 Conclusiones.....	58
CONCLUSIONES.....	59
RECOMENDACIONES.....	60
REFERENCIAS BIBLIOGRAFICAS.....	61
BIBLIOGRAFÍA.....	65
GLOSARIO DE TÉRMINOS.....	69



ÍNDICE DE TABLAS

Tabla 1: Descripción de los actores del negocio.	20
Tabla 2: Descripción de los trabajadores del negocio.	20
Tabla 3: Descripción del Caso de Uso del negocio.	22
Tabla 4: Descripción de los actores del sistema.	28
Tabla 5: Descripción del Caso de Uso del sistema Autenticar usuario.	30
Tabla 6: Descripción del Caso de Uso del sistema Realizar videoconferencia.	33
Tabla 7: Descripción del Caso de Uso del sistema Mostrar videoconferencia bajo demanda.	35
Tabla 8: Caso de prueba del caso de uso: Autenticar usuario.....	56
Tabla 9: Descripción de las variables del caso de prueba: Autenticar usuario.	56
Tabla 10: Descripción de las no conformidades de la aplicación.....	58

ÍNDICE DE ILUSTRACIONES

Fig. 1: Arquitectura de la tecnología Media Streaming	10
Fig. 2: Diagrama de Caso de Uso del negocio.	21
Fig. 3: Diagrama de actividades del negocio.....	23
Fig. 4: Modelo de Objetos.	24
Fig. 5: Diagrama de Casos de Uso del componente audiovisual.....	27
Fig. 6: Diagrama de Casos de Uso de la Teleconsulta de alasMEDIGEN.	28
Fig. 7: Modelo de la Vista Lógica de la arquitectura.	38
Fig. 8: Diagrama de clases del diseño del CU Realizar videoconferencia.	39
Fig. 9: Diagrama de clases del diseño del CU Mostrar videoconferencia bajo demanda.....	40
Fig. 10: Modelo de datos.....	42
Fig. 11: Modelo de clases persistentes.	43
Fig. 12: Diagrama de secuencia del CU Realizar videoconferencia.	44
Fig. 13: Diagrama de despliegue.	48
Fig. 14: Diagrama de componentes del CU Realizar videoconferencia.	49
Fig. 15: Ejemplo del estándar de código.	51
Fig. 16: Ejemplo de prueba de caja blanca.	53
Fig. 17: Grafo del flujo de código para el caso de prueba: Verificar Planificación.....	53

INTRODUCCIÓN

Humanidad y comunicación son dos temas fuertemente relacionados, los lazos que los unen han sido condicionados por la necesidad que ha caracterizado a la humanidad de comunicarse, venciendo barreras, como la distancia. Mediante el desarrollo tecnológico se ha logrado minimizar la influencia de este factor en las comunicaciones, a través de sistemas que permiten transmitir señales digitales. Novedosos avances tecnológicos desarrollados en este campo favorecen todas las esferas en las que el ser humano interactúa. En la actualidad, una de sus aplicaciones más importantes consiste en la asistencia médica a distancia, actividad que se realiza a nivel mundial.

En Cuba se brinda una atención diferenciada a la esfera de la salud, lo que ha dado como resultado el prestigio alcanzado a nivel mundial en este campo. La creación del Programa Nacional para el Diagnóstico y Prevención de Malformaciones Congénitas y Enfermedades Hereditarias, forma parte del conjunto de acciones desarrolladas para obtener estos resultados. Los centros y servicios de genética médica necesitan una coordinación nacional de la actividad de asistencia médica que realizan, para llevar a cabo eficazmente su labor.

La Universidad de las Ciencias Informáticas (UCI), institución educacional y centro de desarrollo de software, ha creado soluciones informáticas para satisfacer las necesidades de informatización del país. Dentro de las cuales se encuentra alasMEDIGEN: “Sistema Informático de Genética Médica”, aplicación que contiene un conjunto de módulos que permiten la integración y gestión de la información de distintos estudios que se realizan en el Centro Nacional de Genética Médica (CNGM). Esto demandó que los especialistas que se encontraban en lugares remotos se pudieran comunicar, para satisfacer esta necesidad se desarrolló la Teleconsulta Genética.

Este módulo ofrece la comunicación entre usuarios de la red, a través de conferencias basadas en texto, las cuales no posibilitan que los genetistas tengan un contacto visual y acústico con los pacientes. Es de vital importancia que los médicos escuchen los diferentes sonidos que estos puedan emitir, pues las características propias de la voz, o los ruidos de una auscultación son imprescindibles para el diagnóstico de algunas enfermedades. En este proceso también influyen los movimientos que los pacientes puedan realizar, ya sea el tipo de marcha o incluso algunos movimientos estereotipados, la forma de mover los ojos, y otras particularidades que permitan a los especialistas realizar un diagnóstico más preciso.

En el año 2009 se desarrolló una investigación que se basó en una propuesta de implementación de nuevos medios de comunicación para la Teleconsulta Genética, donde se descartó la utilización de videoconferencias debido a las características limitadas que presentaba la red de Infomed. Este trabajo

se limitó a proponer el desarrollo de la Teleconsulta Genética utilizando audioconferencias como medio de comunicación, donde no se llegaron a materializar los resultados. (1)

Luego del análisis previo realizado se identificó como **problema de la investigación**: ¿Cómo lograr que la Teleconsulta Genética de alasMEDIGEN permita realizar discusiones de casos a distancia para lograr un diagnóstico más preciso? Teniendo como **objeto de estudio**: la videoconferencia, delimitando el **campo de acción**: la videoconferencia en el sistema alasMEDIGEN.

Para dar solución al problema de la investigación, se identifica como **objetivo general**: Desarrollar un componente audiovisual para el módulo Teleconsulta Genética del sistema alasMEDIGEN.

En correspondencia, se trazan los siguientes **objetivos específicos**:

- ✓ Analizar los sistemas de Teleconsulta desarrollados a nivel mundial.
- ✓ Realizar el análisis y diseño del componente audiovisual para el módulo Teleconsulta Genética del sistema alasMEDIGEN.
- ✓ Implementar el componente.
- ✓ Validar el sistema a través de pruebas de software.

Para el cumplimiento de los objetivos específicos, se proponen las siguientes **tareas de la investigación**:

- ✓ Estudio del estado del arte referente a los sistemas de Teleconsulta.
- ✓ Definición de los requisitos del sistema.
- ✓ Modelación de los casos de uso del sistema.
- ✓ Definición de la arquitectura del componente.
- ✓ Modelación de los diagramas de clases del diseño.
- ✓ Implementación del componente audiovisual para la Teleconsulta Genética.
- ✓ Validación del sistema utilizando las pruebas de software de caja negra y caja blanca.

El trabajo está estructurado en cuatro capítulos, donde se abarca todo lo relacionado con la investigación realizada.

Capítulo 1. Fundamentación teórica: Se realiza un estudio del arte de los diferentes sistemas informáticos de videoconferencias en línea existentes en el mundo, se definen los conceptos más importantes, se exponen las tendencias, tecnologías y herramientas utilizadas para su desarrollo.

Capítulo 2. Características del sistema: Se modela el negocio y se realiza una propuesta del sistema, definiéndose los requisitos funcionales y no funcionales. Se definen los actores, los trabajadores y los casos de uso del sistema.

Capítulo 3. Diseño del sistema: Se modela el sistema para que soporte todos los requisitos y restricciones, se describen la arquitectura y los patrones de diseño que se puedan utilizar como base para la implementación. Se representan los diagramas de clases y el diagrama de despliegue.

Capítulo 4. Implementación y prueba: Muestra la estructura de la implementación del sistema, modelándose los diagramas de componentes. Se define el estándar de codificación utilizado. Además se especifican las pruebas de software realizadas al sistema.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el capítulo se abordan los principales conceptos relacionados con la Telemedicina, centrados en la Teleconsulta, donde la discusión de los usuarios se hace posible mediante las videoconferencias. Luego de un análisis bibliográfico, se exponen además, las herramientas, las metodologías, las tecnologías y el fundamento teórico de los principales componentes utilizados en la difusión de la videoconferencia en la red.

1.1 Teleconsultas.

En la última década se han incorporado cada vez más las TIC en el área de la salud y se han profundizado las experiencias en curso a raíz de la prioridad otorgada a esta nueva plataforma tecnológica. El potencial de estas tecnologías en el sector de la salud debe analizarse en función de los desafíos que dicho sector enfrenta en cada país en particular, buscando optimizar la organización y la calidad de la atención a los pacientes. La Telemedicina es una de las herramientas que se desarrolló con este objetivo. Su empleo trae como consecuencia un aumento en la disponibilidad de recursos y profesionales de la salud altamente capacitados, instalaciones, equipamiento y medicamentos adecuados.

La Organización Mundial de la Salud (OMS), define la Telemedicina como: "el suministro de servicios de atención sanitaria, en cuanto la distancia constituye un factor crítico, por profesionales que apelan a las tecnologías de la información y de la comunicación con objeto de intercambiar datos para hacer diagnósticos, preconizar tratamientos y prevenir enfermedades y heridas, así como para la formación permanente de los profesionales de atención de salud y en actividades de investigación y de evaluación, con el fin de mejorar la salud de las personas y de las comunidades en que viven". (2)

Telemedicina es el uso de las telecomunicaciones para proveer información y servicios médicos. (3)

El concepto sobre este tema abordado por INSALUD¹ es: "La utilización de las tecnologías de la información y de las comunicaciones como un medio de proveer servicios médicos, independientemente de la localización de los que ofrecen el servicio y de los pacientes que lo reciben y el intercambio de la información necesaria para la actividad asistencial". (3)

Por lo antes expuesto, se puede decir que la Telemedicina es una forma de proveer servicios médicos, utilizando para ello las telecomunicaciones. Además de ayudar a la formación de profesionales de la

¹ Instituto Nacional de la Salud, entidad de prestaciones sanitarias públicas de España.

Capítulo 1: Fundamentación Teórica

salud en la investigación científica, teniendo como finalidad un mejor desarrollo de la medicina en general.

La Telemedicina está dividida en servicios de tratamiento, de diagnóstico y de información o educación. El servicio de diagnóstico de forma más detallada, se divide en el Telediagnóstico, la Teleconsulta, el Telemonitoreo, la Telecirugía y la Teleeducación.

"La Teleconsulta es la búsqueda de información médica o asesoramiento por parte de personal médico local o externo, utilizando tecnologías de información y telecomunicación. Esta puede ser desarrollada tanto entre pacientes y profesionales de la salud como entre estos últimos". La Asociación Americana de Telemedicina (ATA) la conceptualizó como: "servicio que usa las telecomunicaciones para presionar datos médicos, los cuales pueden ser audio, fotografía o videos, entre un paciente y un profesional de salud para utilizarla en asistencia, diagnóstico o un plan de tratamiento". (2)

Con ayuda de estos criterios, se concluye que la Teleconsulta permite compartir interactivamente imágenes médicas e información del paciente entre los médicos especialistas ubicados en lugares remotos.

La Teleconsulta se divide en dos clasificaciones según el tipo de comunicación que utilice:

- ✓ Sincrónica
- ✓ Asincrónica.

El envío de información clínica es el soporte fundamental de las Teleconsultas asíncronas. Es la modalidad más usada, donde se utiliza el correo electrónico como medio para el envío de datos. Están basadas en el modo de operación "almacenamiento y envío". Su principal ventaja es que posibilita que las partes involucradas no tengan que estar presentes en la transferencia de la información. Además permite capturar y almacenar imágenes estáticas o en movimiento del paciente, así como audio y texto, que son económicas e ideales para alto volumen de trabajo y pruebas.

La Teleconsulta sincrónica es conocida como el envío de información en tiempo real, que involucra a profesionales de la salud y a los pacientes, apoyada de las videoconferencias como tecnología de mayor impacto. Estas influyen positivamente en el desarrollo de diagnósticos rápidos y aumentan la integración de técnicas adicionales que mejoran la confiabilidad de la información clínica. Aunque es necesario mencionar, que su costo es más elevado. (2)

La videoconferencia realizada a través de una red de comunicaciones, intercambiando audio y video, que permite el enlace entre dos o más puntos separados es considerada como una reunión virtual.

Existen varias modalidades de videoconferencias. En función del número de sitios enlazados:

- ✓ Cuando son dos, se denomina conferencia punto a punto.
- ✓ Cuando son más de dos, se le conoce como multipunto.

Según el tipo de participación:

- ✓ Uno a uno: videoconferencia con dos participantes.
- ✓ Uno a varios: uno de los participantes difunde su información al resto, mientras que estos solo lo escuchan.
- ✓ Varios a varios: videoconferencia con tres o más participantes en la que existe retroalimentación.

El uso de este medio posibilita ahorros en costos de viajes, facilidad para la toma de decisiones rápidas y una comunicación sencilla. La modalidad de videoconferencia que se desarrollará será multipunto de varios a varios. (4)

1.2 Situación actual del módulo Teleconsulta Genética de alasMEDIGEN.

En el proyecto alasMEDIGEN: "Sistema Informático de Genética Médica", desde el 2008, se incorporó el módulo Teleconsulta Genética, que permite coordinar y realizar las consultas a distancia entre el CNGM y los diferentes centros de genética del país, de forma que se puedan servir las peticiones de toda la red nacional de genética médica y gestionar las solicitudes realizadas por los profesionales de la genética del país.

El genetista municipal envía una solicitud al genetista provincial para discutir un caso que se presentó a su consulta y que no se conoce el diagnóstico del mismo. El genetista provincial analiza la información recibida. En caso de que la solicitud no sea aprobada el sistema permite enviar una notificación con la respuesta de la misma. Si procede su discusión, el genetista de la provincia envía una citación a cada uno de los especialistas que participarán en la discusión del caso con los datos del mismo. Luego se realiza la discusión del caso mediante el uso del chat con la interacción médico-médico existiendo el intercambio de imágenes y textos, permitiendo decidir si el mismo quedó resuelto, si se pospone o si se remite para el nivel superior. En todos los casos se guarda el historial de la discusión, un resumen de lo discutido, el diagnóstico, que puede ser definitivo o no, las recomendaciones y los datos de los participantes. (5)

1.3 Antecedentes.

1.3.1 Sistemas de Teleconsulta en el ámbito internacional.

En varios países se han desarrollado potentes sistemas de Teleconsulta que constituyen un recurso significativo utilizado para mejorar la atención médica a la población. A continuación se exponen algunos de los ejemplos más novedosos.

El Hospital Garrahan de Argentina ha instrumentado un programa de Telemedicina mediante el cual se dará soporte a los centros de salud del interior del país, con consultas de alta complejidad. El Instituto Oftalmológico Zaldívar provee interconsultas virtuales de tipo diferido o en tiempo real, específicamente en Buenos Aires, permitiéndoles a los pacientes consultarse con el doctor Zaldívar, director científico de este centro, sin necesidad de viajar al instituto.

En Jamaica, se han utilizado varios proyectos de Telemedicina en diferentes especialidades médicas, como dermatología, oncología, psiquiatría y servicios de medicina domiciliaria, obteniéndose notables resultados.

La Telemedicina rural y en prisiones, así como en el ámbito de especialidades como la Telerradiología, ha jugado un significativo rol en Panamá.

Teleconsulta de Lesiones Pigmentadas: El Departamento de Dermatología Médico-Quirúrgica del Hospital Virgen Macarena de Sevilla, trabaja con un sistema de selección especializada mediante Teleconsulta, para ofrecer un sistema asistencial que abarca una amplia gama de servicios, que cubre desde atención primaria hasta el nivel hospitalario. Es una aplicación que ha demostrado resultados en la medicina, su fundamento está basado en un sistema a tiempo diferido o asincrónico. (6)

TESIS Telemedicina S.L: Empresa perteneciente a la Universidad Politécnica de Madrid. Centra su trabajo en el desarrollo y aplicación de sistemas de Telemedicina y Tecnologías de la Información y la Comunicación para el sector sanitario. Potencializa sistemas destinados a mejorar la calidad del cuidado al paciente, fomenta la equidad, llevando la atención médica a áreas remotas y dando lugar a un mejor control de la gestión sanitaria. Desarrolló un sistema de Teleconsulta con el mismo nombre, donde el médico general se comunica con el especialista y comparten imágenes y diagnósticos del paciente, de manera asincrónica. (7)

Red Universitaria de Telemedicina (RUTE): Desarrollada en Brasil, apoya el perfeccionamiento de la infraestructura para Telemedicina, educación e investigación, y promueve la integración de proyectos entre las instituciones que la integran y la atención de salud, mediante la disponibilidad compartida de fichas médicas, consultas, exámenes y segundas opiniones. La RUTE posee una infraestructura sólida

Capítulo 1: Fundamentación Teórica

que consta de una eficaz y eficiente conexión en ciudades con redes comunitarias de educación e investigación, un núcleo de Telemedicina y Telesalud, sala de videoconferencia, Teleconsulta, Telediagnóstico, personal capacitado en TIC y videoconferencia y grupos de interés específicos en funcionamiento. (8)

Software TM-64: Desarrollado por Real Time Telemedicine Services, S.A. (RT2S), empresa española perteneciente al Grupo Comitas, que mediante la utilización de la Telemedicina y con el objetivo de mejorar la calidad asistencial a los pacientes, facilita la prestación de servicios médicos de forma remota entre organizaciones sanitarias, facultativos y pacientes ubicados en cualquier lugar del mundo. Evita los obstáculos que dificultan o imposibilitan el desplazamiento de las personas y complementa las actuaciones de los facultativos, a los que proporciona el apoyo de un equipo de especialistas. Esta empresa trabaja con la Red Telemática TM-64, que posibilita la monitorización remota y el reconocimiento de pacientes en tiempo real. El software posibilita labores asistenciales logrando compartir experiencias, sesiones clínicas, conferencias o cursos de formación de manera simultánea e interactiva, gracias a un equipamiento de multi-conferencia. (9)

Phemium Consultant: Perteneciente a la empresa Androme Phemium es una plataforma en modo software como servicio, creada y gestionada por ANDROME.es. Diseña, desarrolla y comercializa soluciones de software de alta calidad en las áreas de comunicaciones, aplicaciones de televisión digital y aplicaciones interactivas basadas en la experiencia de usuario. Usa tecnologías de comunicaciones y multimedia, permite la conexión con los usuarios mediante videollamadas e incluye un riguroso control de acceso. (10)

GVOConference: Tecnología perteneciente a la compañía del mismo nombre, que ha buscado ofrecer soluciones de videoconferencia web profesionales, con sólo una cámara web convencional y un micrófono común en cuestión de minutos cualquier persona en cualquier lugar del mundo con conocimientos mínimos en el uso del ordenador podrá estar usando un salón de videoconferencia web. Tiene uso en formato monopunto y multipunto hasta cuatro personas o moderadores al mismo tiempo. La actual tecnología permite hacer presentaciones en PowerPoint, hacer uso del pizarrón, intervención de los asistentes con sus propias cámaras y micrófonos, poner videos precargados o cargados directamente desde de Youtube.com, compartir archivos, grabar las conferencias y otras características. (11)

AXISCameraStation: De la entidad Axis Communications, brinda un servicio de videos en la red, que proporciona funciones de supervisión de video, grabación y gestión de eventos. Los usuarios pueden realizar una grabación de video continua, programada, activada por alarma y/o por detección de

movimiento. El software dispone de múltiples funciones de búsqueda de eventos grabados. Funciona con cámaras de red y codificadores de video de Axis. Estas cámaras utilizadas, tienen una carcasa a prueba de agresiones, incluyen la posibilidad de asignarle una dirección IP (Internet Protocol), codifican la señal, basadas en estándares H.264, y contienen secuencias Motion JPEG. (12)

Estas aplicaciones aunque benefician a los centros de salud donde se aplican, no se pueden integrar o utilizar en el desarrollo de la solución de esta investigación por las siguientes razones:

- ✓ Las aplicaciones TESIS Telemedicina S.L y Teleconsulta de Lesiones Pigmentadas desarrollan las Teleconsultas de manera asíncrona.
- ✓ AXISCameraStation utiliza cámaras IP para enviar los flujos de video y audio, tecnología muy costosa.
- ✓ El sistema de Teleconsulta de RUTE y la aplicación GVOConference necesitan un ancho de banda muy amplio para transmitir las videoconferencias.
- ✓ Los sistemas TM-64 y Phemium Consultant son aplicaciones privativas.

1.3.2 Sistemas de Teleconsulta en el ámbito nacional.

Relacionado con esta temática, en Cuba se ha desarrollado el apoyo a procesos asistenciales de forma diferida en el proyecto de Telemedicina realizado en la región central del país. Implementado sobre la Red de Cardiología y Cirugía Cardiovascular, posibilitando el envío de historias clínicas de los pacientes que serán tratados a los especialistas, para que estos definan la intervención que se les realizará.

En general las líneas que más se han trabajado son las relacionadas con el diagnóstico, específicamente la Imagenología y la Telepatología. También se creó la Red de Telediagnóstico para el Sistema Nacional de Salud. (13)

La Empresa de Telecomunicaciones de Cuba (ETECSA), es la única que ofrece la posibilidad de realizar videoconferencias, solo en determinados lugares. Además las empresas deben pagar un precio muy elevado para consumir este servicio y tienen que acudir a los centros de ETECSA, para desarrollar las reuniones virtuales. (14)

En la facultad 7 de la UCI, se desarrolló un Sistema de Gestión Hospitalaria, que tiene un componente de Teleconsulta, pero la solución obtenida está basada en el uso de un servidor streaming Red5, el cual necesita para su funcionamiento un ancho de banda muy amplio. El proyecto Plataforma Telefónica (PLATEL) desarrollado en la Facultad 2, establece las comunicaciones a través de audioconferencias, pero no permite desarrollar videoconferencias.

1.4 Herramientas, tendencias y tecnologías actuales.

1.4.1 Tecnología Media Streaming.

La tecnología básica utilizada en los sistemas de videoconferencia es la compresión digital de audio y video en tiempo real, emitida a través de redes de telecomunicaciones. El hardware o software que realiza la compresión se llama códec (codificador / decodificador). Con la necesidad del desarrollo de nuevas soluciones y diferentes formatos para perfeccionar la transmisión de la videoconferencia, surge la tecnología Media Streaming.

Esta tecnología permite la difusión de contenidos multimedia. Mientras que el usuario permanece recibiendo los datos streaming que le están enviando, puede reproducir el video sin la necesidad de esperar a que finalice la transmisión, la cual puede ser en dos tiempos de reproducción: en vivo o en video bajo demanda. El video se envía después de haber sido codificado, contribuyendo a la compresión de la información y reduciendo en gran medida el ancho de banda requerido. (15)

En la transmisión en vivo se reproduce en la computadora del usuario, un evento que está ocurriendo en el momento. La transmisión de video bajo demanda no es más que la reproducción de un contenido que ha sido grabado previamente, se encuentra almacenado y disponible para consultarse en cualquier momento. La arquitectura de esta tecnología está compuesta por cuatro partes fundamentales: (16)

- ✓ Captura y codificación.
- ✓ Almacenamiento y servicio.
- ✓ Distribución y entrega.
- ✓ Reproducción.



Fig. 1: Arquitectura de la tecnología Media Streaming

Inicialmente se realiza la captura y codificación mediante algún dispositivo especializado para esto. El audio y el video se digitalizan mediante micrófonos y cámaras. En el momento de captura se codifican los flujos de video y audio para realizar el streaming. Los canales de distribución conectan a los clientes con el servidor. Los clientes reciben el streaming a través de un reproductor, que lo descomprime en formato de audio y video tradicional y finalmente lo reproduce. (16)

Media Streaming distribuye sus funciones en dos componentes:

- ✓ Servidor streaming.
- ✓ Reproductor.

Servidor streaming.

El servidor streaming ofrece servicios para la transmisión simultánea de datos en redes de telecomunicaciones. Optimiza la calidad de la recepción en función del ancho de banda disponible. Procesa los datos multimedia en un corto plazo de tiempo, desarrolla funciones de control interactivas que posibilitan gestionar el video y suministra los servicios de audio y video en modo sincronizado.

A continuación se describen algunos servidores existentes:

Real Networks Helix Server: es el más utilizado en el mundo. Está disponible para múltiples plataformas y se puede encontrar en paquetes que incluyen todas las herramientas necesarias para poner en funcionamiento un sistema de streaming. Su problema radica en que es privativo, y el precio de las instalaciones más avanzadas es elevado, y aunque existe una versión básica gratuita es limitada en varios aspectos, solo es permitida por un año.

QuickTime Streaming Server: surgido del mundo Macintosh Software, es un servidor de libre distribución, disponible para entornos Windows y Unix. La gestión del servidor se efectúa desde una página web. La ausencia de un productor de contenidos asociados como el que presenta el paquete de Real Networks constituye el principal inconveniente, debido a que este componente realiza la codificación de los flujos de video y audio. Aunque se puede destacar que presenta significativos beneficios como: compatibilidad con la mayoría de los clientes disponibles sobre cualquier plataforma y ofrece la posibilidad de reenvío de flujos a otros servidores adicionales. (15)

FluMotion: Es una plataforma de streaming multiformato para publicar contenido de audio y video a través de Internet desarrollado por la empresa española Flumotion Services. Está basada en el framework multimedia GStreamer, escrito en el lenguaje de programación C. Este servidor proporciona un paquete básico de su servidor de streaming bajo la licencia GPL y es utilizado para hacer streaming

bajo demanda. Soporta los formatos Windows Media, MP3 o Flash, así como los estándares abiertos Ogg Vorbis/Ogg Theora y WebM. (17)

VideoLan: es un proyecto libre de código abierto de una plataforma de video, que incluye reproductor, servidor y otros plugins. Algunas de las principales ventajas que presenta se describen a continuación:

- ✓ Puede funcionar tanto como reproductor o como servidor.
- ✓ Centrado en el proyecto VLC, que soporta diferentes tipos de formatos y señales de entrada, como DVD, satélite, sistema de ficheros, televisión digital, terrestre, y otras.
- ✓ Es una aplicación servidora y cliente de flujos de video. Compatible con la mayoría de las plataformas, tales como Windows, Mac OS X, Linux, BeOS, FreeBSD, OpenBSD y Familiar Linux.
- ✓ Tiene soporte de protocolos: RTSP, UDP, RTP, HTTP, FTP.
- ✓ Trabaja con formatos: MPEG, AVI, ASF, WMV, WMA, MP4, MOV, 3GP, OGG, OGM, Annodex, Matroska, WAV, RAW audio, RAW DV, FLAC, FLV.
- ✓ Soporta interfaces gráficas, como Consola, Telnet y Web.
- ✓ Posee plugin para Iceweasel y Firefox.

Se apoya en dos programas fundamentales:

- ✓ VLC Media Player: Puede jugar el papel de servidor y de cliente para hacer y recibir streaming por la red. Hace streaming con todos los archivos que puede leer.
- ✓ VLS (VideoLan Server): Hace streaming de archivos MPEG-1, MPEG-2 y MPEG-4, DVDs, canales digitales de satélite, canales de televisión digital terrestre y videos en vivo. (16)

Por las anteriores características se seleccionó como servidor streaming a utilizar el VideoLan para desarrollar las videoconferencias. Además es necesario utilizar el FluMotion para emitir los videos bajo demanda.

Reproductor.

El reproductor a utilizar será el VLC Media Player, en correspondencia con el servidor streaming seleccionado.

1.4.2 Estándares de codificación para la transmisión de video y audio.

El esquema de compresión, es uno de los aspectos más importantes que emplea Media Streaming. Elimina datos innecesarios para salvar espacio y obtener ventaja en los servicios telemáticos. Para que los archivos de video se puedan enviar a través de la red, de manera más eficiente, es necesario que

estos sean codificados, para reducir y eliminar los datos redundantes, sin afectar la calidad de la imagen, en ello influye el nivel de compresión de la técnica que se utilice.

Los códecs posibilitan codificar el flujo de información para su transmisión, almacenamiento, cifrado y recuperación, descifrando los datos del mismo modo para la reproducción o manipulación en un mejor formato.

La Unión Internacional de Telecomunicaciones (UIT) ha propuesto varios estándares de codificación de los flujos de video y audio, que favorecen la transmisión de videoconferencias, con el objetivo de lograr compatibilidad e interoperabilidad en las soluciones.

El estándar H.323 define una familia de protocolos para satisfacer los requerimientos técnicos en los servicios de comunicaciones entre redes basadas en paquetes, que pueden o no ofrecer calidad de servicio. Establece un amplio conjunto de características y funciones necesarias para las transmisiones en redes. Define los componentes que debe tener un sistema de videoconferencia. Especifica los protocolos que gestionan la preparación, establecimiento, control de estado, mensajería, códec de audio/video, transferencia de datos y fin de llamada o conexión. A continuación se describen estos protocolos: (16)

- ✓ H.245: es el estándar de control y señalización que es utilizado para la apertura y cierre de canales lógicos y el intercambio de información sobre la capacidad de transmisión y recepción de medios de los terminales.
- ✓ H.225: es donde se encuentran definidos los mensajes de señalización, definen la manera en que se gestionan los datos, video, audio e información de control de redes basadas en la conmutación de paquetes. Registra los terminales en una zona H.323, para el control de admisión de puntos finales y la notificación de cambios de estado de la conexión o en el ancho de banda disponible.

El estándar que se utiliza en esta investigación es el H.323, debido a que es el que aplica el servidor streaming VideoLan.

1.4.3 Metodología de desarrollo.

Proceso Unificado de Rational (RUP)

Las metodologías son un conjunto de procedimientos, técnicas y ayudas a la documentación para realizar productos de software. Representan un pilar importante en los procesos de desarrollo de software, debido a que minimizan las pérdidas de tiempo en la generación de software. Es una guía que indica el qué hacer y el cómo actuar cuando se busca obtener buenos resultados en una

investigación determinada. Para el desarrollo del sistema se definió utilizar la metodología RUP, que es un proceso de desarrollo de software dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental. RUP es el resultado de estudios e investigaciones realizadas por tres desarrolladores Ivar Jacobson, Grady Booch, James Rumbaugh. Para preparar los esquemas de un sistema de software utiliza el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés). En su modelación define como sus principales elementos: los trabajadores, las actividades, los artefactos y el flujo de actividades. (18)

Consta de cuatro fases:

- ✓ Inicio
- ✓ Elaboración
- ✓ Construcción
- ✓ Transición

Además se divide en diferentes flujos de trabajo:

- ✓ Modelación del negocio.
- ✓ Requerimientos.
- ✓ Análisis y diseño.
- ✓ Implementación.
- ✓ Prueba.
- ✓ Instalación.
- ✓ Administración de configuración y cambios.
- ✓ Administración del proyecto.
- ✓ Ambiente.

1.4.4 Lenguaje de Modelado.

El Lenguaje Unificado de Modelado es utilizado para especificar y documentar los artefactos de un software. Es útil para el modelado completo de sistemas complejos. Abarca temas conceptuales y concretos, entiéndase como estos: los procesos del negocio, funciones del sistema, esquemas de base de datos y componentes de software reutilizables. Cuenta con varios tipos de diagramas que visualizan diferentes aspectos de lo que se quiere representar. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. (19)

1.4.5 Lenguajes de programación utilizados.

C++

Es un lenguaje de programación orientado a objetos, se suele decir que es híbrido o multiparadigma, debido a que permite también la programación estructurada. Cuando se compila genera un código nativo con un alto grado de optimización en memoria y velocidad. Otra de sus características más importantes es que permite la libre utilización de punteros por parte de los programadores. Es responsabilidad del programador la administración de la memoria, por lo que la asignación y liberación de esta, se puede definir a conveniencia. En cuanto a la sobrecarga de operadores, posibilita la redefinición de los mismos, es decir, que a sus operadores propios se les puede redefinir la semántica. En la presente investigación, se utiliza este lenguaje para lograr una compatibilidad con la biblioteca del VLC, desarrollada en C++. Dicha biblioteca permite desarrollar un servidor streaming que implemente los componentes de una videoconferencia, según el estándar H.323. (20)

PHP 5.

El Hypertext Preprocessor (PHP), es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Una de sus características más potentes es su soporte para gran cantidad de bases de datos, tales como InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL, entre otras. Ofrece la integración con varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa desde generar documentos en formato PDF hasta analizar código XML. (21)

1.4.6 Frameworks de desarrollo.

Los frameworks son soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución). Es una estructura compuesta de componentes personalizables e intercambiables para la implementación de una aplicación. (22)

QT4.

QT es un framework multiplataforma, que se utiliza para el desarrollo de aplicaciones, está escrito en C++. Actualmente existe una amplia documentación sobre este tema. Integra varios componentes, simplifica el proceso de construcción de proyectos y soporta plugins. Tiene componentes propios que permiten implementar la arquitectura Cliente/Servidor e interactuar con bases de datos de forma muy sencilla. Este framework se utilizará en el desarrollo del componente audiovisual. (23)

Symfony 1.0.22

Es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja.

Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Es compatible con la mayoría de los gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas Unix, Linux, como en plataformas Windows. Otro aspecto importante es que para la implementación del sistema alasMEDIGEN se utilizó este framework en la versión 1.0.22, por lo cual es necesario utilizarlo para la correcta integración de la solución al sistema. (24)

1.4.7 Herramienta CASE.

Visual Paradigm 6.4

Herramienta CASE que permite modelar el sistema, que abarca todas las actividades y artefactos propuestos por la metodología de desarrollo de software. Se puede trabajar tanto en la ingeniería directa como en la inversa. Posibilita generar código y documentación. Facilita el proceso de revisión de las aplicaciones. Es multiplataforma (GNU/Linux). Brinda un lenguaje común facilitando la comunicación y la agilidad dentro del desarrollo. Tiene permanente sincronía durante la realización del software entre modelo y código. Permite integrarse con diferentes entornos de desarrollo integrado (IDE, por sus siglas en inglés). (25)

1.4.8 IDE utilizados.

Qt Creator 4.7

Qt Creator es un IDE para el desarrollo de aplicaciones multiplataforma. Se caracteriza por aprovechar toda la potencialidad del lenguaje C++, el cual brinda la posibilidad de tener soporte para la refactorización de código, también posee una interfaz gráfica de usuario (GUI) y ayuda integrada, diseñador de formularios, un depurador visual, además de una herramienta de proyecto y administración. (26)

Netbeans 7.0

NetBeans es un IDE que provee una estructura para los proyectos que se pueden crear, propone un esqueleto para organizar el código fuente, el editor conjuntamente integra lenguajes como: HTML, JavaScript y CSS. Es un proyecto de código abierto con una gran base de usuarios y una comunidad en constante crecimiento. Consta de un sistema de detección de errores de sintaxis en tiempo real. Permite la integración de varios frameworks como Symfony, Hibernate y otros. (27)

1.4.9 Sistema Gestor de Base de Datos (SGBD).

MySQL 5.1

MySQL es un SGBD relacional, multihilo y multiusuario, además de ser el más popular y utilizado a la hora de desarrollar páginas Web dinámicas y sitios de comercio electrónico. Se suele trabajar en combinación con PHP, y comparte con este las siguientes características que lo convierten en una elección segura:

- ✓ **Gratuito:** Se trata de software libre que puede ser utilizado sin limitación alguna.
- ✓ **Popularidad:** Son innumerables las páginas donde se puede encontrar información.
- ✓ **Rapidez:** La velocidad de proceso de MySQL es legendaria.
- ✓ **Versatilidad:** Trabaja tanto con sistemas operativos basados en Unix como con el sistema operativo Windows, de Microsoft.
- ✓ **Seguridad:** Un sistema de privilegios y contraseñas que es muy flexible y seguro. Las contraseñas son seguras, ya que su tráfico está encriptado cuando se conecta con un servidor.

MySQL se encuentra disponible bajo una doble licencia: GPL y Comercial, estando esta última destinada a empresas que quieran utilizarlo en sus desarrollos, sin la necesidad de liberar sus trabajos para las redes víricas de la MySQL. Se utiliza en el desarrollo de esta solución porque es el SGBD definido por el marco regulatorio de Softel para las aplicaciones de salud. (28)

1.4.10 Servidor de aplicaciones Web.

Apache es la plataforma de servidores web de código fuente abierto más poderosa del mundo. Apache Server 2.0 es una solución web más flexible, transportable, segura, escalable y con mayor rendimiento. Está diseñado para ser un servidor web que pueda funcionar en la más amplia variedad de plataformas y entornos a través de su diseño modular. Este diseño permite a los administradores de sitios web elegir las características que van a ser incluidas en el servidor, seleccionando los módulos que se van a cargar, ya sea al compilar o al ejecutar el servidor. (29)

1.4.11 Patrones arquitectónicos utilizados.

Los patrones arquitectónicos especifican un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes.

Cliente-Servidor (C/S): Es un patrón de arquitectura que consiste básicamente en un cliente que realiza peticiones a otro programa (servidor) que le da respuesta. Permite centralizar la gestión de la información y separar las responsabilidades de cada programa, esto facilita y clarifica el diseño del sistema.

La red de comunicaciones cliente-servidor permite que todos los clientes estén conectados a un

Capítulo 1: Fundamentación Teórica

servidor, en el cual se centralizan los diversos recursos y este los pone a disposición de los clientes cada vez que estos son solicitados.

C/S es una relación entre procesos corriendo en máquinas separadas. El servidor (S) es un proveedor de servicios. El cliente (C) es un consumidor.

International Business Machines (IBM) define al modelo Cliente/Servidor como: "Tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o clientes, resultan en un trabajo realizado por otros computadores llamados servidores". (30)

Sus principales ventajas son:

- ✓ Centralización del control: Los accesos, recursos y la integridad de los datos son controlados por el servidor, de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos.
- ✓ Escalabilidad: Se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado en cualquier momento, o se pueden añadir nuevos nodos a la red.
- ✓ Fácil mantenimiento: Al estar distribuidas las funciones y las responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio. Esta independencia de los cambios también se conoce como encapsulación.

La interfaz Socket es una API para redes TCP/IP que permite la comunicación entre aplicaciones C/S. Para conectar dos programas en una red, es necesario que cada uno utilice la interfaz antes mencionada. Se establece una conexión entre los Socket de cada programa, se leen y se escriben los datos necesarios y una vez terminados los procesos se cierra la conexión. (30)

Modelo Vista Controlador (MVC): Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. En Symfony, las partes del MVC presentan las siguientes características:

El modelo: Solo se encarga del acceso a los datos almacenados en el gestor de base de datos. Ha sido dividido en dos capas, la capa de acceso a los datos y en la capa de abstracción de la base de datos.

La vista: Las páginas web suelen contener elementos que se muestran de forma idéntica a lo largo de toda la aplicación: cabeceras de la página, el layout genérico, el pie de página y la navegación global. En la mayor parte de las veces sólo cambia el interior de la página. Por este motivo, la vista se separa en un layout y en una plantilla. Normalmente, el layout es global en toda la aplicación o al menos en un grupo de páginas. La plantilla sólo se encarga de visualizar las variables definidas en el controlador.

El controlador: Una parte importante de su trabajo es común a todos los controladores de la aplicación. Entre las tareas comunes se encuentran el manejo de las peticiones del usuario, el manejo de la seguridad, cargar la configuración de la aplicación y otras tareas similares. Por este motivo, el controlador se ha dividido en un controlador frontal, que se encarga de realizar las tareas comunes y las acciones, que incluyen el código específico del controlador de cada página. (31)

Modelo/Vista: Este patrón es aplicado por el framework QT4 que maneja la relación entre los datos y la forma en que se presenta al usuario. Modelo/Vista es una adaptación del patrón arquitectónico MVC, está constituido por dos capas, en una se agrupa la vista y el controlador y la otra capa es el modelo. Esto ofrece a los desarrolladores una mayor flexibilidad para personalizar la presentación de los elementos. Proporciona una interfaz estándar del modelo que permite una amplia gama de fuentes de datos, separando la manera en que la información es almacenada, con la forma en que se presentan al usuario. (32)

1.5 Conclusiones.

Con la realización de este capítulo se abordaron conceptos fundamentales relacionados con el problema planteado. También se realizó un estudio sobre los sistemas existentes que presentan videoconferencias en línea, las cuales permiten la realización de reuniones en tiempo real, determinando que ninguno cumplía con las características requeridas. Se definió la utilización de RUP como metodología de desarrollo, Visual Paradigm 6.4 como herramienta de modelado, C++ y PHP 5 como lenguajes de programación, Symfony 1.0.22 y QT4 como frameworks de desarrollo, Netbeans 7.0 y Qt Creator 4.7 como entornos de desarrollo integrado. Además se describió la tecnología Media Streaming la cual es utilizada para la transmisión de video y audio en la red. Se definieron los patrones arquitectónicos a utilizar, para lograr un diseño y una implementación con la calidad requerida.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.

En el presente capítulo se describe el negocio actual para comprender el objetivo del sistema, se especifican los actores y trabajadores del negocio así como los casos de uso identificados. Se representa el diagrama de casos de uso, el diagrama de actividades y el modelo de objetos del negocio. Se realiza el levantamiento de requisitos. Se especifican los actores y los casos de uso del sistema, así como la descripción textual de los mismos. Se obtiene una concepción general del sistema a implementar, por medio de la metodología RUP y UML como lenguaje de modelado.

2.1 Modelado del negocio.

2.1.1 Descripción del modelo del negocio.

Los trabajadores de Salud Genética que pertenecen al Programa Nacional para el Diagnóstico y Prevención de Malformaciones Congénitas y Enfermedades Hereditarias, están ubicados en centros distribuidos por todo el país. Debido a la existencia de casos específicos se requiere la atención de especialistas, los que no se encuentran en todas las zonas del país, dificultando así la asistencia médica especializada a los pacientes.

2.1.2 Actores y trabajadores del negocio.

Actor	Descripción
Solicitante	Interactúa directamente con el negocio, es el que solicita al especialista del nivel superior que se discuta un caso determinado que no se le haya encontrado diagnóstico, o por ser una patología que requiera de la intervención de un equipo multidisciplinario a nivel terciario de salud, o para la realización de monitoreos especiales.

Tabla 1: Descripción de los actores del negocio.

Trabajador	Descripción
Especialista Superior.	Representa al especialista provincial y al nacional. Es el encargado de analizar la información de un caso a discutir y aprobarlo. Planifica la discusión de un caso y emite el diagnóstico del mismo.

Tabla 2: Descripción de los trabajadores del negocio.

2.1.3 Caso de Uso del negocio.

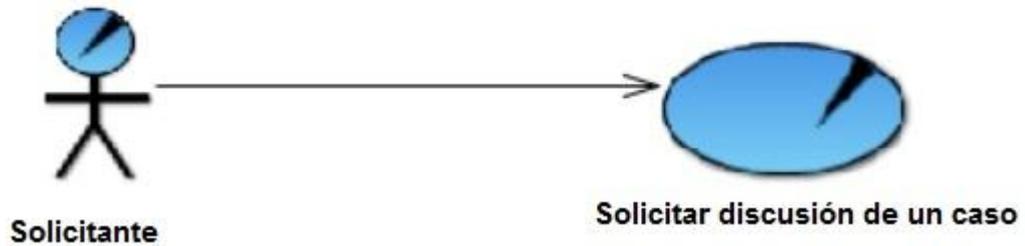


Fig. 2: Diagrama de Caso de Uso del negocio.

2.1.4 Descripción textual del Caso de Uso del negocio.

Caso de Uso del negocio	Solicitar discusión de un caso
Actores	Solicitante (inicia)
Resumen	El caso de uso se inicia cuando el Solicitante solicita la discusión de un caso determinado al Especialista Superior y le envía la información del caso. Este último analiza la solicitud, si considera necesaria la discusión del caso la planifica y se realiza la discusión del caso. Luego emite el diagnóstico correspondiente. El Solicitante recibe el diagnóstico, finalizando así el caso de uso.
Acción del actor	Respuesta del proceso de negocio
1. El Solicitante solicita la discusión de un caso y envía la información del mismo.	2. El Especialista Superior analiza la solicitud recibida. 3. El Especialista Superior si considera necesario realizar la discusión del caso, la planifica. 4. El Especialista Superior realiza la discusión del caso con otros especialistas. 5. El Especialista Superior emite el diagnóstico.
6. El Solicitante recibe el diagnóstico, finalizando así el	

caso de uso.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	3.1 El Especialista Superior, no considera necesario realizar la discusión del caso. (Ir al paso 5 del flujo normal)

Tabla 3: Descripción del Caso de Uso del negocio.

2.1.5 Diagrama de actividades del negocio.

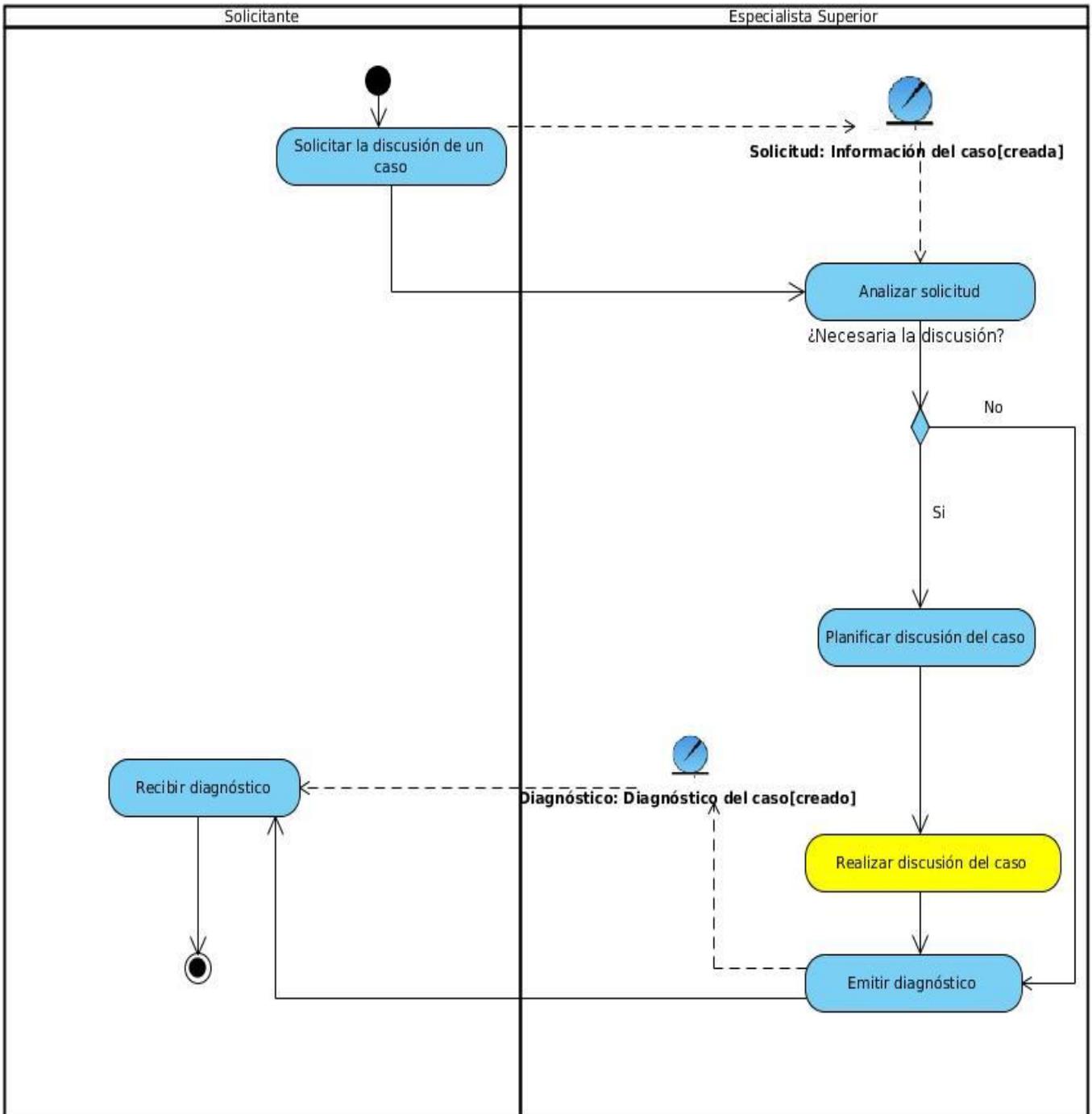


Fig. 3: Diagrama de actividades del negocio.

2.1.6 Modelo de Objetos.

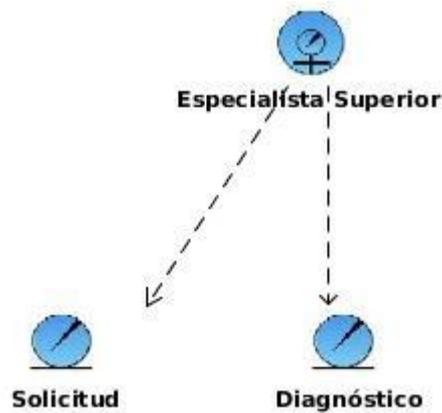


Fig. 4: Modelo de Objetos.

2.2 Propuesta del sistema.

El sistema consiste en un componente audiovisual que permite la realización de las Teleconsultas, a través de videoconferencias. En las videoconferencias pueden participar un máximo de cuatro especialistas. El componente está integrado por dos aplicaciones de escritorio: Cliente y Servidor. La aplicación Cliente será utilizada por los especialistas que participen en las discusiones de los casos, para enviar los flujos de video y audio hacia la aplicación Servidor. Esta última permite crear y guardar el mosaico de la videoconferencia, donde se integran todos los flujos de video y audio recibidos de los usuarios. Luego la aplicación Servidor envía el mosaico a los usuarios, quienes lo visualizan a través de la aplicación Cliente. Además se incorpora una nueva funcionalidad al módulo de Teleconsulta Genética del sistema alasMEDIGEN, que permite a los usuarios buscar y visualizar las videoconferencias almacenadas en las que hayan participado.

2.3 Levantamiento de requisitos.

Como guía del desarrollo del sistema se especifican y se describen los requisitos del software, además de lograr un acuerdo entre los clientes y el equipo de desarrollo en cuanto a lo que la aplicación debe hacer.

2.3.1 Requisitos funcionales.

Los requisitos funcionales (RF) son capacidades o condiciones que el sistema debe cumplir. Estos no alteran la funcionalidad del producto, es decir, que se mantienen invariables sin importar las propiedades o cualidades que se relacionen. (33)

- ✓ RF1- Autenticar usuario.

- ✓ RF2-Enviar flujo de video y audio.
- ✓ RF3-Capturar flujo de video y audio.
- ✓ RF4-Realizar videoconferencia.
- ✓ RF5- Guardar videoconferencia.
- ✓ RF6- Terminar videoconferencia.
- ✓ RF7- Buscar videoconferencia.
- ✓ RF8- Visualizar videoconferencia.

2.3.2 Requisitos no funcionales:

Los requisitos no funcionales (RNF) son propiedades o cualidades que el producto debe tener. Los mismos forman una parte significativa de la especificación del sistema.

RNF1- Requerimientos de apariencia o interfaz externa.

La interfaz de usuario será sencilla, amigable e intuitiva, de fácil navegación, no contendrá numerosas imágenes para evitar demoras en la respuesta de cualquier acción del usuario. Además la interfaz estará diseñada para una óptima visualización.

RNF2- Usabilidad.

Se garantizará un acceso fácil y rápido a los usuarios. El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de una computadora.

RNF3- Seguridad.

La información estará protegida contra accesos no autorizados, utilizando mecanismos de validación que puedan garantizar el cumplimiento de usuario y contraseña, de manera que solo las personas permitidas tengan acceso a la aplicación garantizando así la confidencialidad. Se utiliza para esto el componente de seguridad SAAA basado en el modelo de Autenticación, Autorización y Auditoría del Registro de Integración de Salud (RIS).

RNF4- Requisitos legales, de derecho de autor y otros.

La UCI posee todos los derechos sobre la aplicación. Las herramientas y las tecnologías en que está basado el proyecto deberán cumplir con las licencias de software GNU/GPL.

RNF5- Confiabilidad.

La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción. El sistema estará protegido contra el acceso de usuarios no autorizados.

RNF6- Interfaz.

Interfaces de usuario

Las ventanas del sistema contendrán claro y bien estructurados los datos, además de permitir la interpretación correcta de la información. La entrada de datos incorrecta será detectada e informada al usuario. Todos los textos y mensajes en pantalla aparecerán en idioma español. El diseño de la interfaz del sistema responderá a la ejecución de acciones de una manera rápida, minimizando los pasos a dar en cada proceso. La interfaz será sencilla, amigable, intuitiva y de fácil navegación por el usuario.

Interfaces de comunicación

Para el intercambio electrónico de datos entre aplicaciones se usarán los protocolos TCP/IP y HTTP.

- ✓ HTTP: Protocolo de transferencia de hipertexto orientado a transacciones siguiendo el esquema solicitud-respuesta entre un cliente y un servidor. Es utilizado para enviar los flujos de video entre el cliente y el servidor. (34)
- ✓ TCP/IP: Permite una conexión de la computadora cliente con el servidor streaming.

RNF7- Hardware.

La infraestructura de Hardware deberá permitir aumentar la cantidad de servidores o adicionar componentes de hardware en función de disminuir la carga, sin que sea necesario realizar modificaciones al software. Para el desarrollo y ejecución de la aplicación propuesta se necesita como mínimo:

Para el servidor de aplicación:

- ✓ Pentium 4 a 2.5 GHz o superior.
- ✓ 1 GB RAM o superior.
- ✓ Tarjeta de video de 512MB o superior.
- ✓ 500 GB de espacio libre en Disco Duro.
- ✓ Tarjeta de red (MODEM o red con TCP-IP).
- ✓ 1024 Kb/s de ancho de banda disponible.

Para las PC de los clientes:

- ✓ Pentium 4 a 2.5 GHz o superior.
- ✓ 512 MB RAM o superior.
- ✓ Tarjeta de red (MODEM o red con TCP-IP para conexión al servidor).
- ✓ Tarjeta de video de 512MB o superior.

- ✓ Micrófono.
- ✓ 256 Kb/s de ancho de banda disponible.
- ✓ Además es necesario contar con una webcam y un micrófono para poder capturar el flujo de video y audio.

RNF8-Software.

Se requiere disponer para el funcionamiento del sistema de un servidor que cuente con Sistema Operativo Linux, servidor Apache 2.0 o superior y MySQL 5 o una versión superior. El puerto seleccionado para la comunicación entre las aplicaciones Cliente y Servidor fue el 5421, debido a que se encuentra en el rango de puertos registrados para realizar estas funciones. Es necesario que este puerto esté liberado. En las PC de los usuarios debe de estar instalado el driver Video4Linux (V4L). Además es necesario contar con las siguientes librerías instaladas:

- ✓ LibVLC 1.1.12 o superior.
- ✓ Libavcodec.

RNF9- Rendimiento.

Los tiempos de respuestas y velocidad de procesamiento de la información deben ser rápidos, no mayores de 10 segundos.

2.4 Diagrama de Casos de Uso del sistema.

Un diagrama de casos de uso del sistema representa gráficamente a las funcionalidades y su interacción con los actores. Cada caso de uso debe comunicarse con al menos un actor. Los elementos que pueden aparecer en un diagrama de casos de uso son: actores, casos de uso y relaciones entre ellos. El mismo constituye una entrada para el análisis, el diseño y las pruebas.

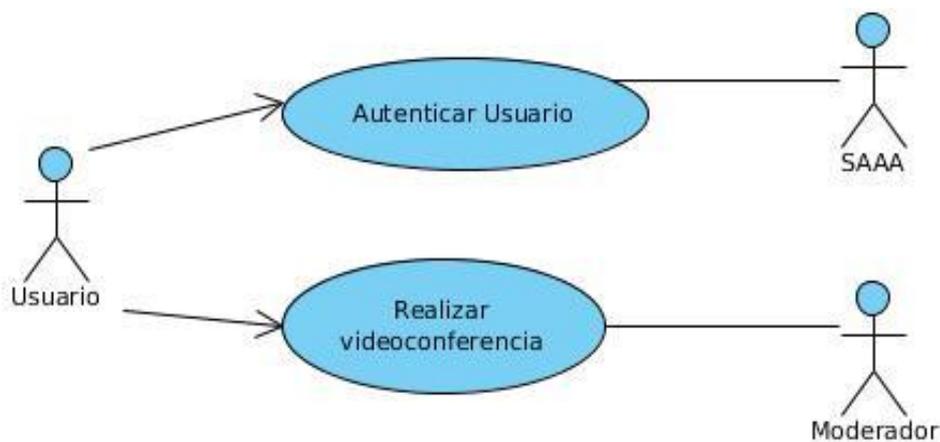


Fig. 5: Diagrama de Casos de Uso del componente audiovisual.

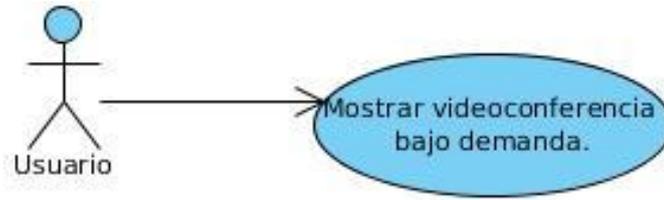


Fig. 6: Diagrama de Casos de Uso de la Teleconsulta de alasMEDIGEN.

2.4.1 Descripción de los actores del sistema.

Actor	Descripción
Usuario	Representa a los especialistas municipales, provinciales y nacionales. Accede a la aplicación Cliente para participar en la discusión de un caso a través de una videoconferencia.
Moderador	Se encarga de iniciar y terminar la videoconferencia de una discusión de un caso, a través de la aplicación Servidor.
SAAA	Representa el componente de seguridad SAAA que pertenece al RIS.

Tabla 4: Descripción de los actores del sistema.

2.4.2 Descripción de los Casos de Uso del sistema.

Caso de Uso (CU) 1	Autenticar usuario
Actor	Usuario
Descripción	El CU se inicia cuando el Usuario accede a la <i>interfaz Registrarse en el sistema</i> , introduce su usuario y contraseña. El sistema verifica los datos y finaliza cuando el sistema muestra la <i>interfaz Configuración de la conexión</i> .
Referencia	RF1
CU asociados:	Ninguno.
Precondiciones:	
<i>Interfaz Registrarse en el sistema</i>	



Descripción	<p>A: Campo de texto para introducir el usuario.</p> <p>B: Campo de texto para introducir la contraseña.</p> <p>C: Botón para registrarse en el sistema.</p>
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1. El Usuario accede a la <i>interfaz Registrarse en el sistema</i>, introduce el nombre y su contraseña y selecciona la opción Iniciar Sesión.</p>	<p>2. El sistema verifica que estén correctos los datos introducidos a través del componente de seguridad SAAA y muestra la <i>interfaz Configuración de la conexión</i>, finalizando así el caso de uso.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>2.1 Si el Usuario no introduce los datos correctamente el sistema muestra un mensaje de error: “Usuario o contraseña no válidos”. (Vuelve al paso 1 del flujo normal).</p> <p>2.2 En caso de que el RIS no esté disponible se muestra un mensaje de error: “El RIS no está disponible”. (Vuelve al paso 1 del flujo normal).</p>

Capítulo 2: Características del Sistema

Poscondiciones:	
Prioridad:	Crítico.

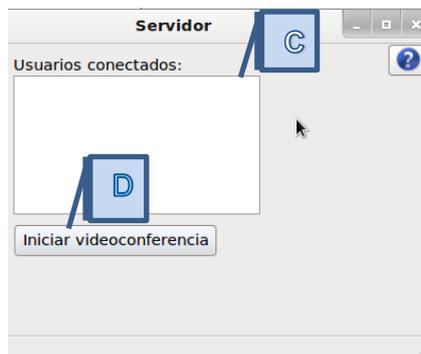
Tabla 5: Descripción del Caso de Uso del sistema Autenticar usuario.

Caso de Uso 2	Realizar videoconferencia
Actor	Usuario (inicia), Moderador.
Descripción	El CU se inicia cuando el Usuario accede a la <i>interfaz Configuración de la conexión</i> de la aplicación Cliente, se conecta a la aplicación Servidor, para ello introduce la dirección IP del servidor. El sistema verifica que el Usuario sea participante de la videoconferencia planificada. La aplicación Cliente comienza a emitir los flujos de video y audio. En la aplicación Servidor el Moderador accede a la <i>interfaz Servidor</i> , selecciona la opción Iniciar, el sistema crea el mosaico de la videoconferencia, conformado por cada uno de los flujos de video y audio recibidos y lo envía a los Usuarios. El Usuario en la aplicación Cliente, en la <i>interfaz Visualizar videoconferencia</i> , selecciona la opción Mostrar videoconferencia. El caso de uso finaliza cuando el Usuario decide concluir la videoconferencia.
Referencia	RF2, RF3, RF4, RF5, RF6
CU asociados:	
Precondiciones:	<ul style="list-style-type: none"> ✓ El Usuario debe estar autenticado en el sistema. ✓ La videoconferencia debe estar planificada en el sistema y el usuario debe ser uno de los participantes de la misma. ✓ Debe existir más de un Usuario conectado en el Servidor y uno de ellos ser el solicitante de la videoconferencia.

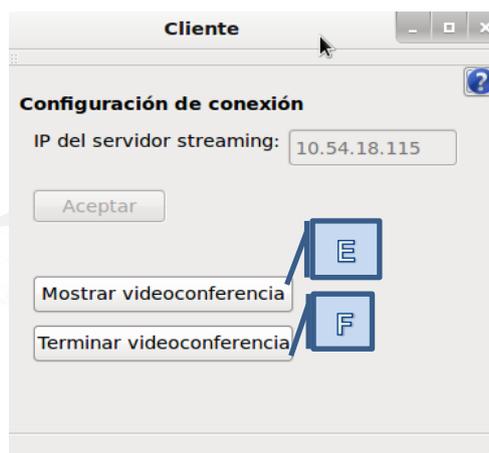
Interfaz Configuración de la conexión



Interfaz Servidor



Interfaz Visualizar videoconferencia



Descripción

A: Campo de texto para introducir la dirección IP del servidor streaming.

Capítulo 2: Características del Sistema

	<p>B: Botón para establecer la conexión con el servidor.</p> <p>C: Campo donde se muestran los usuarios conectados.</p> <p>D: Botón para comenzar la videoconferencia.</p> <p>E: Botón para mostrar la videoconferencia.</p> <p>F: Botón para terminar la videoconferencia.</p>
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Usuario accede a la <i>interfaz Configuración de la conexión</i> de la aplicación Cliente, introduce la dirección IP del servidor y selecciona la opción Aceptar.	2. La aplicación Cliente se conecta a la aplicación Servidor, donde se verifica que el Usuario sea participante de la videoconferencia planificada y se comienzan a emitir los flujos de video y audio.
3. El Moderador accede a la <i>interfaz Servidor</i> y selecciona la opción Iniciar.	4. La aplicación Servidor construye el mosaico de la videoconferencia, conformado por cada uno de los flujos de video y audio recibidos y lo envía a los Usuarios.
5. El Usuario accede a la <i>interfaz Visualizar videoconferencia</i> en la aplicación Cliente y selecciona la opción Mostrar videoconferencia.	7. La aplicación Cliente captura la videoconferencia y la visualiza.
8. El Usuario accede a la <i>interfaz Visualizar videoconferencia</i> de la aplicación Cliente, selecciona la opción Terminar videoconferencia, finalizando así el caso de uso.	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1. Si el Usuario introduce un IP del servidor incorrecto se

Capítulo 2: Características del Sistema

	<p>muestra el mensaje de error: “Error en la comunicación con el Servidor”. (Vuelve al paso 1 del flujo normal).</p> <p>2.2. Si el IP introducido por el Usuario coincide con la dirección del servidor pero la aplicación no se está ejecutando se muestra un mensaje de error: “La conexión fue rechazada. El Servidor está desconectado”. (Vuelve al paso 1 del flujo normal).</p> <p>2.3. Si el Usuario no es participante de la videoconferencia planificada, el sistema lo desconecta. (Vuelve al paso 1 del flujo normal).</p>
	<p>4.1. Si el Moderador accede a la <i>interfaz Servidor</i>, selecciona la opción Iniciar y hay menos de dos usuarios conectados el sistema muestra el mensaje de error: “Por favor, verifique la cantidad de clientes conectados”. (Vuelve al paso 3 del flujo normal).</p> <p>4.2. Si el Moderador accede a la <i>interfaz Servidor</i>, selecciona la opción Iniciar y el solicitante de la videoconferencia no se ha conectado el sistema muestra el mensaje de error: “El solicitante de la videoconferencia no se ha conectado”. (Vuelve al paso 3 del flujo normal).</p>
Poscondiciones:	La videoconferencia quedará almacenada en la PC Servidor y se guardarán los datos de la misma.
Prioridad:	Crítico.

Tabla 6: Descripción del Caso de Uso del sistema Realizar videoconferencia.

Caso de Uso 3	Mostrar videoconferencia bajo demanda.
Actor	Usuario
Descripción	El CU se inicia cuando el Usuario accede al módulo de Teleconsulta del sistema alasMEDIGEN, selecciona la opción “Buscar videoconferencia” e introduce los datos para filtrar la búsqueda. Luego el Usuario selecciona la opción Buscar, para mostrar todos los registros de las videoconferencias

Capítulo 2: Características del Sistema

	almacenadas y debe dar clic en el ícono correspondiente para reproducir el video. El usuario decide terminar la reproducción de la videoconferencia, finalizando así el caso de uso.
Referencia	RF7, RF8
CU asociados:	
Precondiciones:	El Usuario debe estar autenticado en el sistema y ser participante de la videoconferencia que desee buscar.

Interfaz Buscar videoconferencia

The screenshot shows the 'alás MEDIGEN' web application interface. The header includes the logo, user name 'Marta de Jesus Gonzalez Mollinedo', and system information. The left sidebar contains a navigation menu with categories like 'Datos Primarios', 'Historia Clínica Genética', 'Gemelo', 'Discapacidad Intelectual', 'Discapacidad Física', 'Anomalías Cromosómicas', and 'Enfermedades Comunes'. The main content area is titled 'Buscar - Videoconferencia' and contains a search form with fields for 'Nombre del solicitante' and 'Fecha de la videoconferencia', a 'Buscar' button, and a table of search results. Callouts A, B, C, and D point to the search fields, the search button, and the play icons in the results table, respectively.

No. Video	Solicitante	Fecha de la Videoconferencia	
14	Marta de Gonzalez Mollinedo	2012-06-7--22:00	▶
15	Marta de Gonzalez Mollinedo	2012-06-7--22:03	▶
16	Marta de Gonzalez Mollinedo	2012-06-7--22:08	▶
1	Marta de Gonzalez Mollinedo	2012-06-5--18:54	▶
17	Marta de Gonzalez Mollinedo	2012-06-7--22:10	▶
2	Marta de Gonzalez Mollinedo	2012-06-7--11:55	▶
18	Marta de Gonzalez Mollinedo	2012-06-7--22:15	▶
3	Marta de Gonzalez Mollinedo	2012-06-7--11:57	▶
4	Marta de Gonzalez Mollinedo	2012-06-7--11:58	▶
5	Marta de Gonzalez Mollinedo	2012-06-7--11:59	▶

Descripción	<p>A: Campo de texto para introducir el nombre del solicitante de la videoconferencia.</p> <p>B: Botón para seleccionar la fecha de la videoconferencia.</p> <p>C: Botón para buscar la videoconferencia.</p> <p>D: Íconos para reproducir la videoconferencia.</p>
-------------	---

Flujo Normal de Eventos

Capítulo 2: Características del Sistema

Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción Buscar videoconferencia del módulo Teleconsulta Genética en el sistema alasMEDIGEN.	2. El sistema muestra la interfaz Buscar videoconferencias.
3. El Usuario accede a la <i>interfaz Buscar videoconferencia</i> , introduce el nombre del solicitante y la fecha de la videoconferencia que desee visualizar. Luego selecciona la opción Buscar.	4. El sistema muestra un listado de las videoconferencias que cumplan con los criterios de búsqueda.
5. El Usuario da clic en el ícono que aparece al lado de la descripción de la videoconferencia que desee visualizar.	6. El sistema muestra la videoconferencia.
7. El usuario cierra la reproducción de la videoconferencia, finalizando así el caso de uso.	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 Si el Usuario introduce una fecha posterior a la actual el sistema muestra el mensaje de error: "Rango de fecha incorrecto". (Vuelve al paso 3 del flujo normal).
Poscondiciones:	
Prioridad:	Crítico.

Tabla 7: Descripción del Caso de Uso del sistema Mostrar videoconferencia bajo demanda.

2.5 Conclusiones.

En este capítulo se realizó el modelado del negocio y el levantamiento de requisitos para definir las funcionalidades del sistema. Se explicó el negocio en el que se enmarca el sistema. Se realizó el diagrama de caso de uso del negocio, el diagrama de actividades y el modelo de objetos,

Capítulo 2: Características del Sistema

identificándose los actores y los trabajadores. Se definieron ocho funcionalidades y tres Casos de Uso del sistema, así como una descripción de los mismos para una mejor comprensión gráfica y visual de la aplicación.



CAPÍTULO 3: DISEÑO DEL SISTEMA.

El diseño del sistema se define como el proceso de aplicar técnicas y principios con el propósito de definir y modelar un proceso o una aplicación, con suficientes detalles como para permitir su interpretación y realización física. Se realiza para transformar los requisitos de un cliente en un producto o software finalizado, fomentando la calidad en la ingeniería del mismo. Permite desarrollar una arquitectura estable. (35)

En este capítulo se definen las clases del diseño. Se explican los patrones de diseño utilizados y se describe la arquitectura del componente. Se representan los diagramas de secuencias y el diagrama de despliegue.

3.1 Arquitectura.

La arquitectura de un software se basa en una serie de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software. Es un entramado de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos, permite integrar una amplia gama de productos y servicios informáticos. (36)

3.2 Vista Lógica de la arquitectura.

La arquitectura analiza a la aplicación desde varios puntos de vista. En RUP, esta se compone de cinco vistas: Vista Lógica, Vista de Procesos, Vista de Implementación, Vista de Despliegue y estas cuatro regidas por la Vista de Casos de Uso. La Vista Lógica tributa a la etapa de análisis y diseño. La arquitectura lógica apoya principalmente los requisitos funcionales, lo que el sistema debe brindar en términos de servicios a sus usuarios. (37)



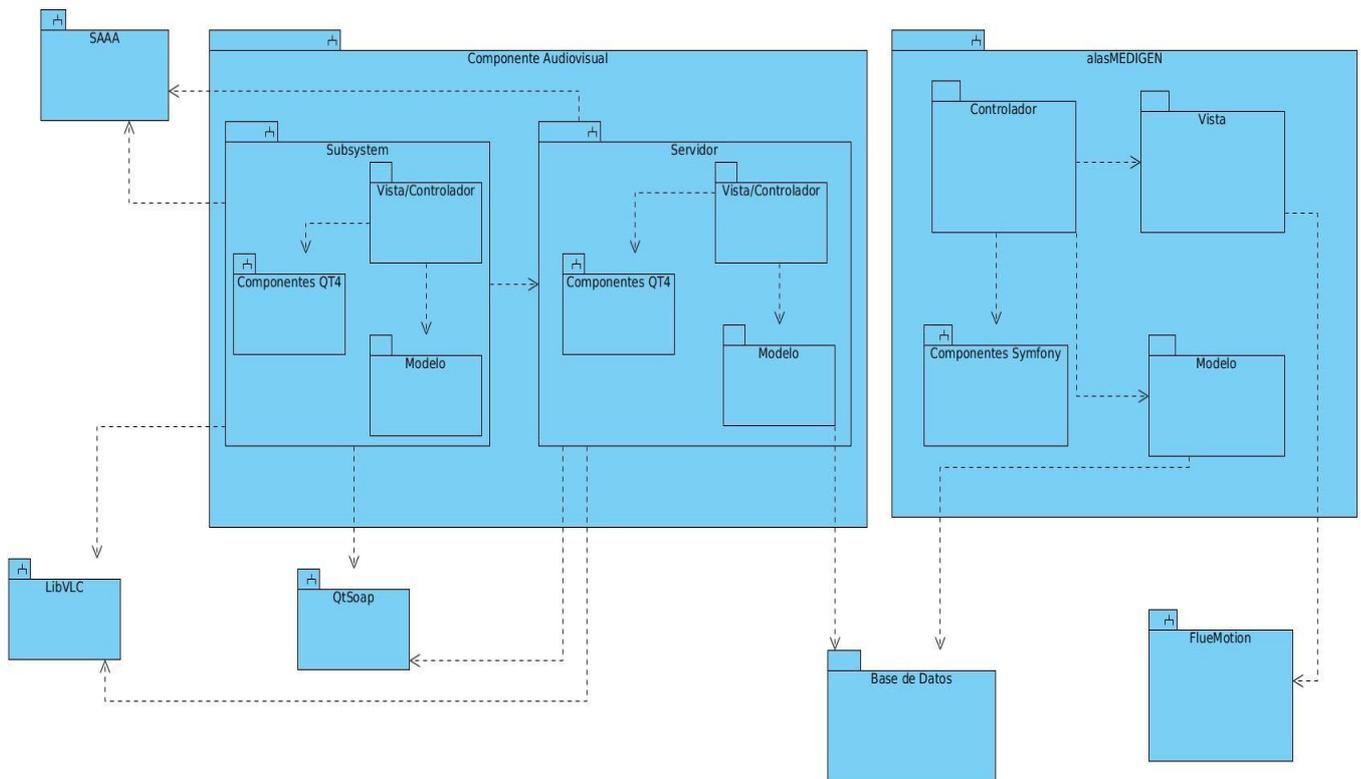


Fig. 7: Modelo de la Vista Lógica de la arquitectura.

La Vista Lógica de la arquitectura que se muestra en la Fig. 7 está integrada por varios subsistemas. El subsistema principal es el *Componente audiovisual* que está integrado por los subsistemas: *Cliente* y *Servidor*, donde se aplica el patrón arquitectónico Cliente-Servidor. En cada uno de estos subsistemas se aplica el patrón arquitectónico Modelo/Vista.

El subsistema *Cliente* representa a la aplicación denominada Cliente que permite a los usuarios conectarse a la aplicación Servidor para enviarle los flujos de video y audio. El subsistema *Servidor* modela a la aplicación con el mismo nombre, que recibe los flujos enviados por cada usuario, crea el mosaico de la videoconferencia, lo guarda y lo envía a cada aplicación Cliente para su visualización.

El subsistema *alasMEDIGEN* representa a la aplicación con el mismo nombre, que aplica el patrón arquitectónico MVC. El subsistema *SAAA* representa al componente de seguridad del RIS, utilizado para la autenticación de los usuarios en el sistema.

El subsistema *Flumotion* representa al servidor streaming Flumotion que interactúa con la capa de la vista del subsistema *alasMEDIGEN*. Este servidor recibe la solicitud con el nombre de la videoconferencia que el usuario desee visualizar y envía el flujo de video para que sea mostrado en el sistema *alasMEDIGEN*.

El subsistema *LibVLC* representa a la biblioteca LibVLC que se utiliza para aprovechar las complejas funcionalidades implementadas por el proyecto VideoLan. Esta librería permite desarrollar un servidor streaming. (38)

El subsistema *QtSoap* representa a la biblioteca con el mismo nombre que se utiliza para el envío de mensajes SOAP, el desarrollo, la depuración o la exploración de los servicios web mediante la versión 1.1 del protocolo SOAP. (23)

3.3 Diagramas de Clases del Diseño.

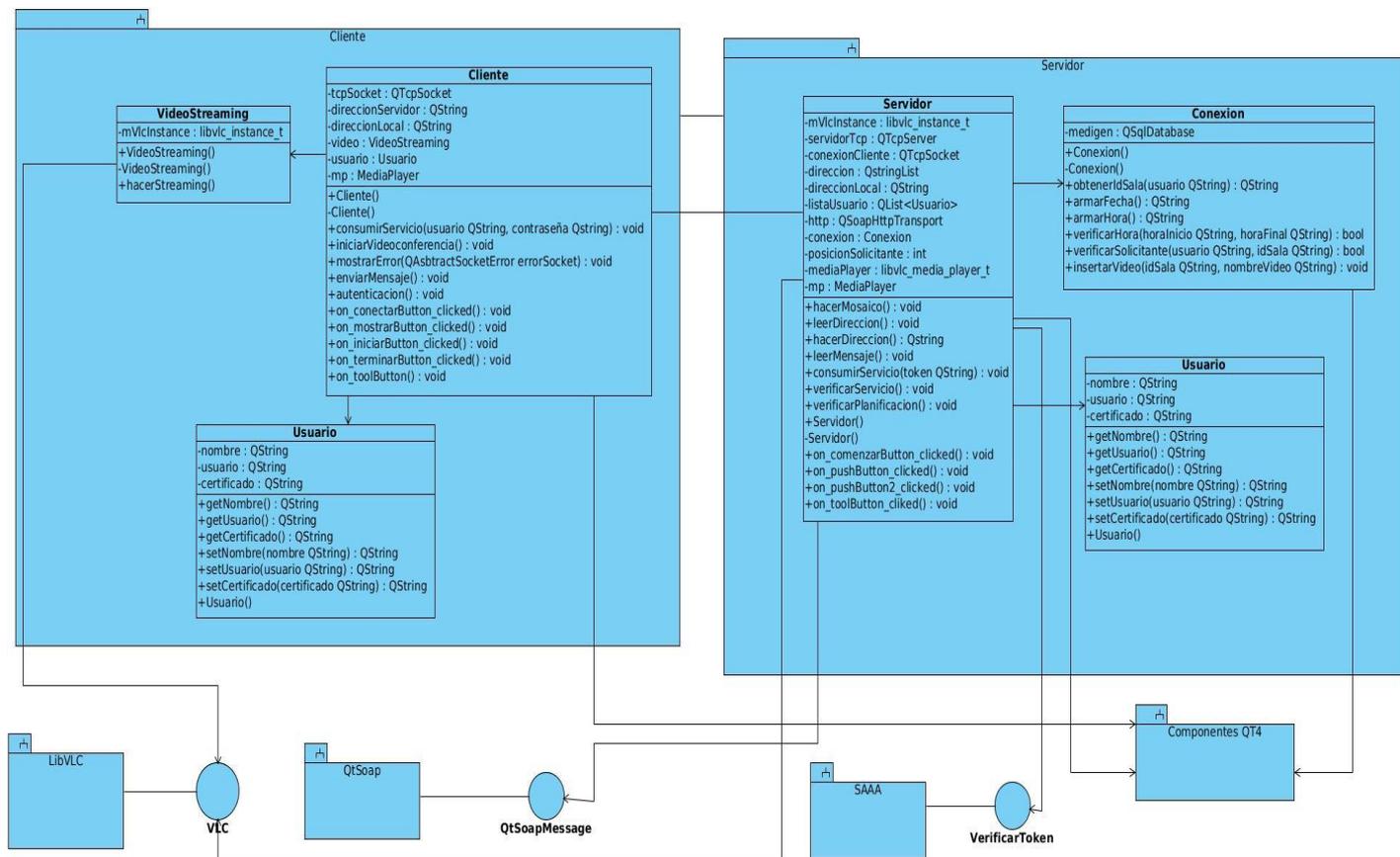


Fig. 8: Diagrama de clases del diseño del CU Realizar videoconferencia.

En el diagrama de clases del diseño del CU Realizar videoconferencia que se muestra en la Fig. 8 se representan dos subsistemas principales: *Cliente* y *Servidor* que se comunican utilizando los componentes del framework QT4 a través de la API Socket. En el subsistema *Cliente* existe una clase controladora denominada *Cliente*, que accede a la clase *VideoStreaming* para interactuar con la interfaz *VLC* de la biblioteca *LibVLC*, que permite enviar los flujos de video y audio hacia el subsistema *Servidor*.

En el subsistema *Servidor* existe una clase controladora nombrada *Servidor*, que recibe los datos del *Cliente* para verificar la seguridad del sistema. Esta comprueba la información del usuario a través de la función *VerificarToken* del componente *SAAA*, utilizando la interfaz *QtSoapMessage* de la biblioteca *QtSoap*. Luego mediante la clase *Conexion* se verifica si el usuario es participante de alguna videoconferencia planificada. Si se cumple esta condición se registran los datos del usuario, se accede a la interfaz *VLC* para crear y enviar el mosaico de la videoconferencia a cada uno de los usuarios conectados. Luego los datos de la videoconferencia se guardan en la base de datos de *alasMEDIGEN* y se almacena el mosaico en el servidor.

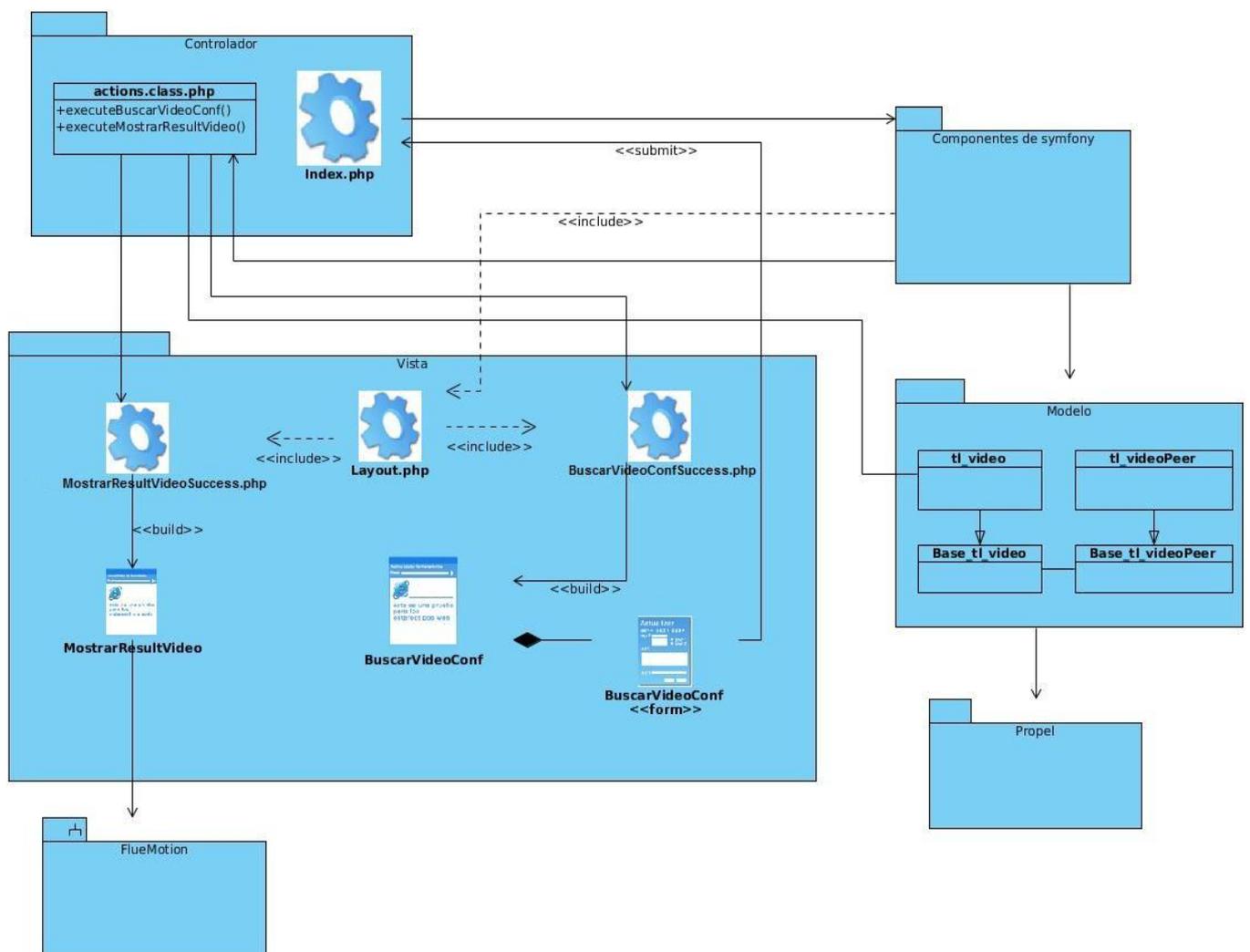


Fig. 9: Diagrama de clases del diseño del CU Mostrar videoconferencia bajo demanda.

En el diagrama de clases del diseño del CU Mostrar videoconferencia bajo demanda que se muestra en la Fig. 9, se representan las relaciones entre las clases que implementan esta funcionalidad. Cuando se selecciona la opción Buscar videoconferencia del módulo Teleconsulta Genética se llama al

controlador frontal *Index.php*. Este redirecciona la petición a la acción *executeBuscarVideoConf* de la clase controladora *actions.class.php*. Esta clase se comunica con la clase *BuscarVideoConfSuccess.php* que construye la vista *BuscarVideoConf*. Esta vista contiene un formulario donde se especifican los criterios de búsqueda de la videoconferencia que se desee visualizar, enviando una petición con la información a la clase *Index.php*. Esta clase redirecciona la petición a la acción *executeMostrarResultVideo* de la clase *actions.class.php* a través de los componentes de *Symfony*. En esta acción se accede a la capa del modelo para buscar los datos de las videoconferencias que cumplan con los criterios de búsqueda y en las cuales el usuario haya participado. El resultado de la búsqueda se muestra en la página *MostrarResultVideo*, construida por la clase *MostrarResultVideoSuccess.php*. Luego se selecciona la videoconferencia que se desee visualizar y se realiza una petición al servidor *Flumotion*, el cual envía el flujo de video correspondiente para visualizarlo.

Para consultar los restantes diagramas de clases del diseño remitirse al expediente de proyecto, específicamente a la plantilla Modelo del diseño_v1.0.

3.4 Modelo de datos

Un modelo de datos es una colección de conceptos bien definidos matemáticamente que ayudan a expresar las propiedades estáticas y dinámicas de una aplicación con un uso de datos intensivo. (39)

alásMEDIGEN tiene una base de datos central donde se guarda la información de los módulos que lo integran. En el desarrollo del componente audiovisual se incorpora una nueva tabla nombrada *tl_video* y se hace una modificación a la tabla *tl_sala_conferencia* de esta base de datos. Esto permite almacenar la información pertinente a las videoconferencias realizadas. En la Fig. 10 se muestra la tabla *tl_video*, que guarda el identificador y el nombre de cada videoconferencia, así como el identificador de la sala de conferencia a la que pertenece. La tabla *tl_sala_conferencia* registra el identificador de la solicitud que fue realizada y a su vez la tabla *tl_solicitud* tiene el identificador del médico que la realizó. Los datos del médico se guardan en la tabla *dg_medico*. Cada médico es un participante. Varios participantes pueden participar en una misma sala de conferencia y en una sala de conferencia pueden interactuar varios participantes.

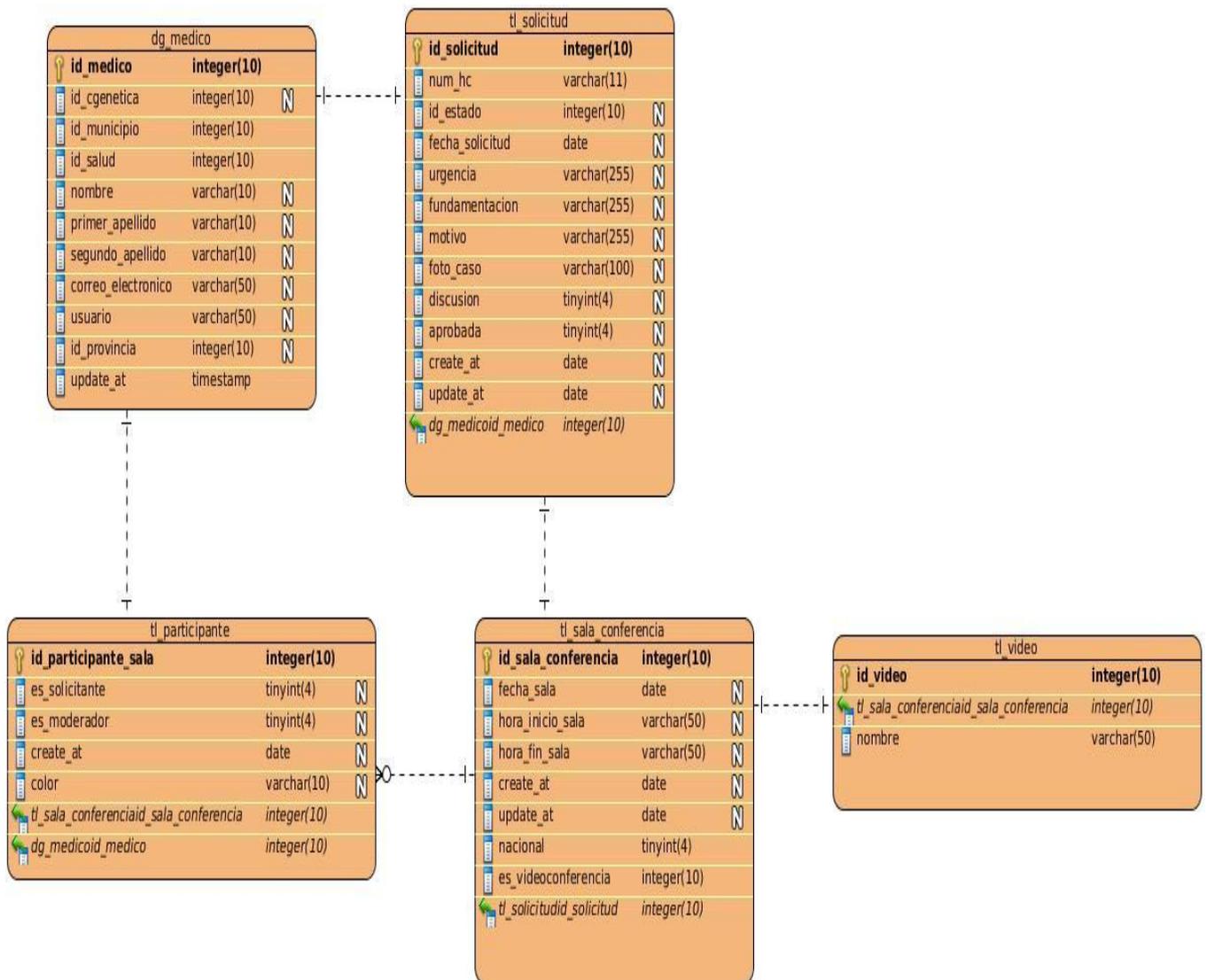


Fig. 10: Modelo de datos.

3.5 Modelo de clases persistentes.

En las aplicaciones que son desarrolladas utilizando Symfony, el acceso y la modificación de los datos almacenados en la base de datos se realizan mediante objetos, para no acceder de forma explícita a la misma. Se utiliza Propel como interfaz para el mapeo objeto-relacional (ORM por sus siglas en inglés) que posibilita acceder a los datos mediante objetos. Esto permite la reutilización de los métodos de los objetos en cualquier parte de la aplicación y encapsula la lógica de los datos. Para traducir el modelo relacional de la base de datos a un modelo de objetos de datos, el ORM necesita la descripción del modelo relacional que se guarda en el archivo schema.yml. Es necesario ejecutar el comando propel-build-model, que analiza el esquema, genera las clases bases del modelo y las guarda en el directorio

lib/model/om de la aplicación. Además genera las clases propias del modelo de datos en el directorio lib/model/. Estas clases propias heredan de las clases bases. Por cada tabla del modelo relacional se generan dos clases bases y dos clases propias del modelo. Cada vez que se modifique la base de datos se modifica el archivo *schema.yml*, teniendo que regenerarse las clases bases del modelo, es por esto que las funcionalidades se deben agregar en las clases propias y no en las clases bases. (40)

En la Fig. 11 se representan las clases del modelo utilizadas. Por cada tabla representada en el modelo de datos anterior se generan cuatro clases. De las veinte clases generadas diez son propias del modelo y las otras diez son las clases bases. Las clases propias tienen las funcionalidades de acceso a los datos con las cuales interactúa la clase *actions.class* del módulo de Teleconsulta de alasMEDIGEN.

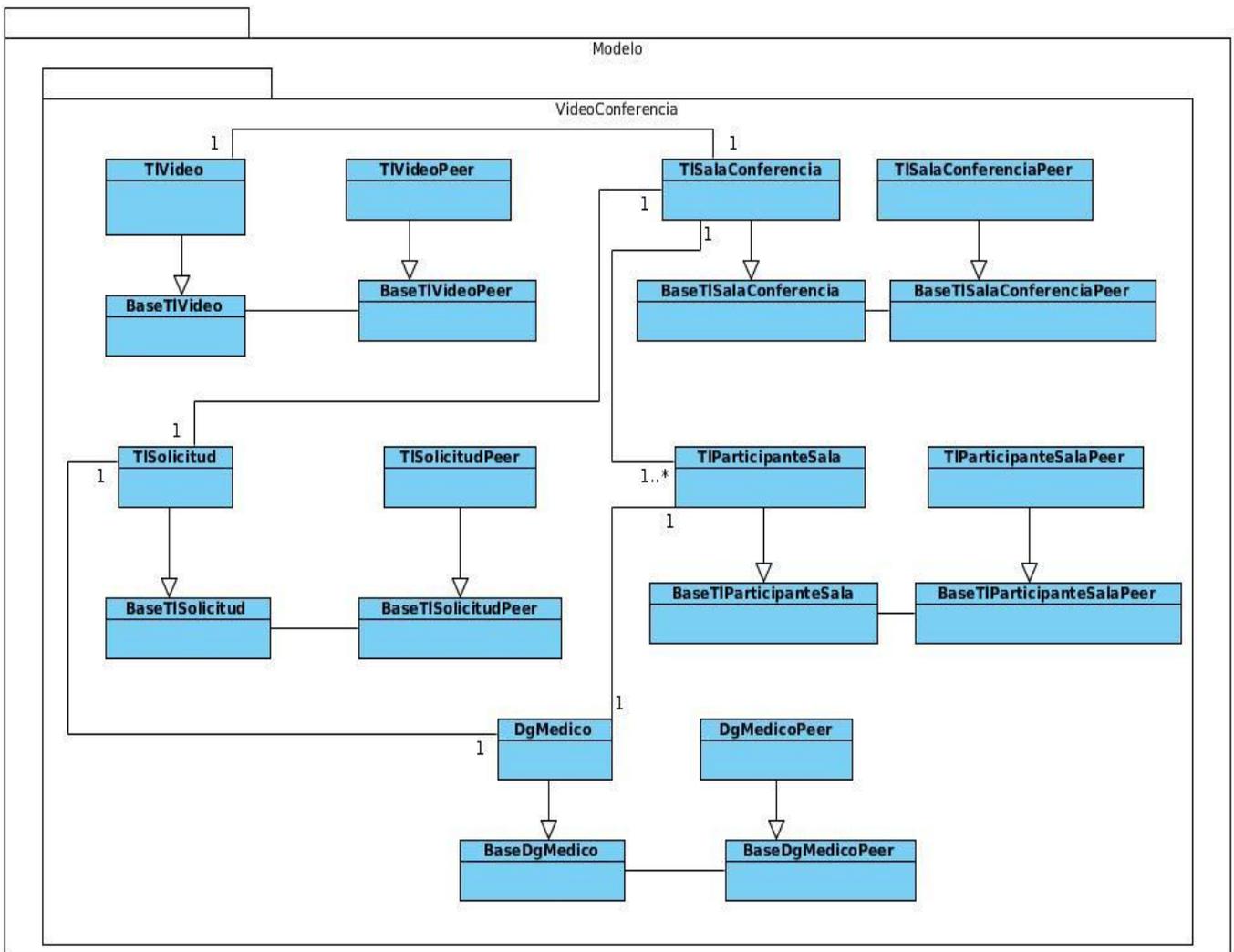


Fig. 11: Modelo de clases persistentes.

3.6 Diagramas de secuencias.

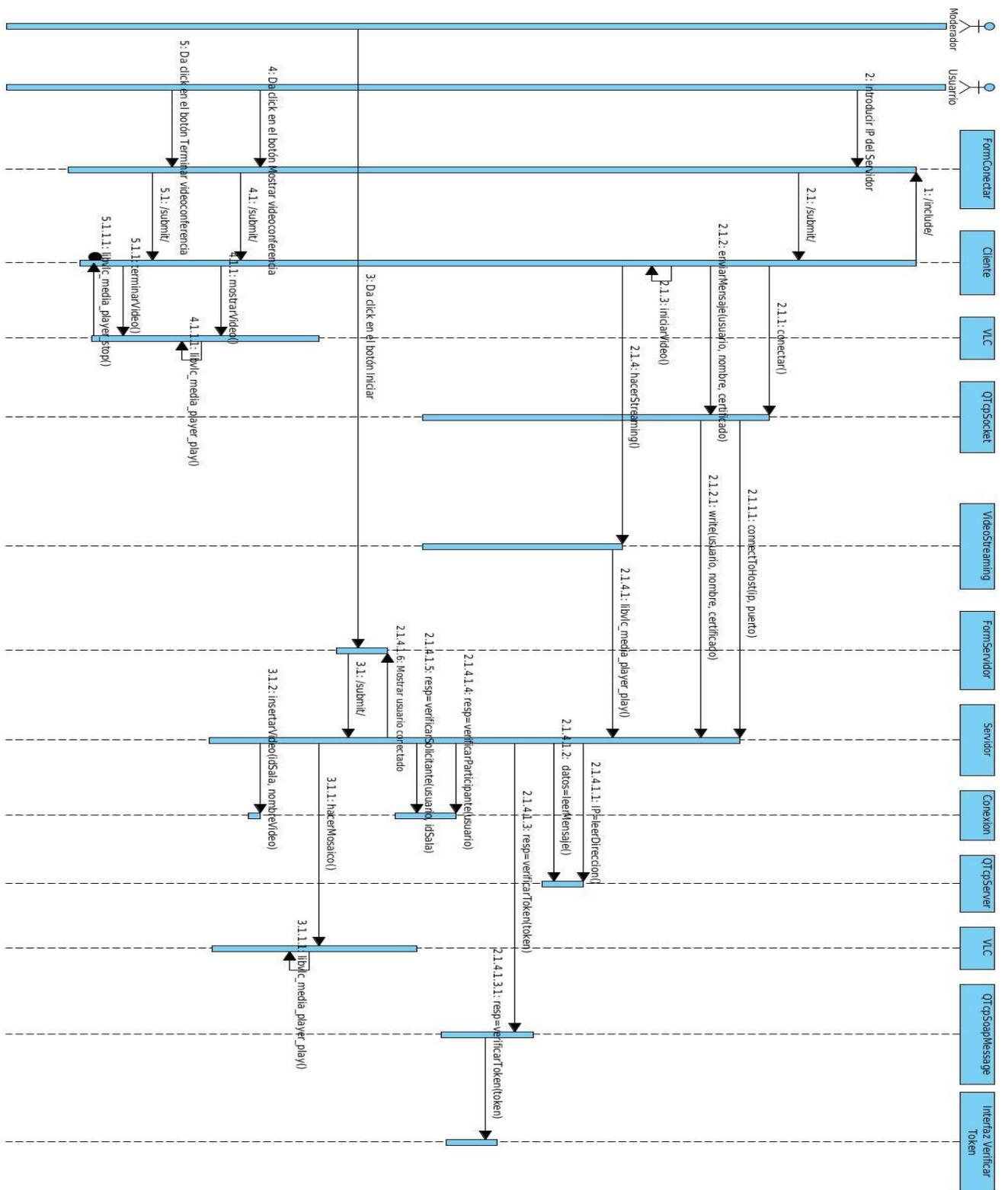


Fig. 12: Diagrama de secuencia del CU Realizar videoconferencia.

En el diagrama de secuencia del CU Realizar videoconferencia que se muestra en la Fig. 12, el Usuario accede al formulario *FormConectar* de la aplicación Cliente, introduce la dirección IP del servidor y selecciona la opción Aceptar. El formulario le envía una petición a la clase *Cliente*, donde se ejecuta la acción *conectar*, pasándole la dirección IP a la funcionalidad *connectToHost* de la clase *QTcpSocket*, encargada de conectar la aplicación Cliente con la aplicación Servidor. Una vez establecida la conexión, a través de la función *enviarMensaje* de la clase *Cliente*, se envía un mensaje a la clase *Servidor* con el usuario, el nombre y el certificado utilizando el método *write* de la clase *QTcpSocket*. Luego se llama a la acción *iniciarVideo* de la clase *Cliente*, para ejecutar la función *hacerStreaming* de la clase *VideoStreaming*, que invoca la acción *libvlc_media_player_play* para enviar los flujos de video y audio hacia la clase *Servidor*.

La clase *Servidor* recibe la notificación de una nueva comunicación por Socket y en la función *leerDireccion* guarda el IP del cliente conectado. La clase *Servidor* ejecuta la función *leerMensaje* donde recibe los datos del usuario, los cuales son enviados a la clase *QtSoapMessage* en la función *verificarToken*. En esta clase se accede a la interfaz *VerificaToken* para consumir el servicio con el mismo nombre. Luego se obtiene una respuesta y se envía la misma a la clase *Servidor*. Una vez verificado esto, la clase *Servidor* ejecuta la función *verificarParticipante*, enviándole los datos del usuario a la clase *Conexion* para comprobar si el usuario está en una videoconferencia planificada. Si se cumple esta condición entonces se invoca la función *verificarSolicitante*, donde se envía la información del usuario a la clase *Conexion* para conocer si el usuario es el solicitante de la videoconferencia y se guarda la posición que le corresponde en el mosaico de la videoconferencia. Posteriormente se muestra el usuario conectado en el formulario *FormServidor*.

El Moderador en la aplicación Servidor accede al formulario *FormServidor*, selecciona la opción Iniciar y se envía una petición a la clase *Servidor* llamando a la función *hacerMosaico*. Esta función utiliza la operación *libvlc_media_player_play* de la interfaz *VLC* que pertenece a la biblioteca *LibVLC* para construir y guardar el mosaico de la videoconferencia. En la clase *Servidor* se accede a la clase *Conexion* para guardar los datos de la videoconferencia mediante la función *insertarVideo*.

El Usuario en la aplicación Cliente accede al formulario *FormConectar*, selecciona la opción Mostrar videoconferencia y se envía una petición a la clase *Cliente* llamando a la función *mostrarVideo* para ejecutar la acción *libvlc_media_player_play* de la interfaz *VLC* y se muestra la videoconferencia.

El usuario accede al formulario *FormConectar*, selecciona la opción Terminar videoconferencia y se envía la petición a la clase *Cliente* ejecutando la función *terminarVideo* que accede a la acción *libvlc_media_player_stop* de la interfaz *VLC*. Esta última acción finaliza la videoconferencia.

Para consultar los restantes diagramas de secuencias remitirse al expediente de proyecto, específicamente a la plantilla Modelo del diseño_v1.0.

3.7 Patrones de diseño.

Un patrón es la descripción de un problema y su solución, que recibe un nombre y que puede emplearse en otros contextos. Indica la manera de utilizarlo en diversas situaciones. Los patrones intentan codificar el conocimiento y los principios ya existentes. El diseño fue elaborado siguiendo patrones basados en la experiencia, que de manera general constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. En este caso se emplearon los patrones de asignación de responsabilidad (GRASP), los que describen los principios fundamentales de la asignación de responsabilidades a objetos y los Gang of four (GOF). (41)

3.7.1 Patrones de asignación de responsabilidades.

Experto: Es el principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearla. Permitiendo mantener el encapsulamiento y la cohesión. En la aplicación Servidor se pone de manifiesto este patrón debido a que existen varias clases, estas tienen sus responsabilidades específicas haciéndolas fáciles de entender y mantener. Esto se evidencia en la clase *Conexion*, que es la encargada de gestionar la conexión a la base de datos, así como todas las consultas que se realizan para obtener información de la misma.

Creador: Este patrón es el encargado de crear las instancias de los objetos para acceder a las entidades que brindan la información necesaria para realizar cualquier acción. En la clase *Cliente* se evidencia la utilización de este patrón, donde se crean instancias de las clases *Usuario* y *VideoStreaming*.

Controlador: Existen otros patrones capaces de crear instancias de otras clases, que ofrecen determinada información que poseen solo ellos y que permiten que cada parte del sistema tenga sus propias responsabilidades independientes. Para lograr la unidad entre todas las partes y que el sistema funcione como un todo es necesario este patrón controlador, que se encargue de unir todas las partes. La aplicación Cliente tiene una clase *Cliente* que hace función de controlador. Igualmente, la aplicación Servidor tiene una clase controladora *Servidor*.

Alta Cohesión y bajo acoplamiento: Symfony permite una excelente organización del trabajo en cuanto a la estructura y responsabilidades bien definidas para todos sus componentes, para que cada uno se centre en las tareas asignadas. Este patrón se evidencia en la forma como el controlador frontal delega las funcionalidades a la clase *actions.class*, esta última se encarga de definir las diferentes acciones a

realizar, asignando a cada una de ellas un objetivo específico. Es decir cada parte se encarga únicamente de sus responsabilidades y se relacionan entre ellas para satisfacer necesidades del sistema en general manteniendo una alta cohesión.

3.7.2 Patrones GOF

Decorator (Envoltorio): Añade funcionalidad a una clase dinámicamente. Por ejemplo, en alasMEDIGEN existe un archivo *layout.php*, que también se denomina plantilla global, donde se almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el *layout*.

Facade (Fachada): Proporciona una interfaz unificada para un conjunto de interfaces de un subsistema. Define una interfaz de alto nivel que hace que el subsistema sea más fácil de usar. Tanto en la aplicación Cliente como en la Servidor se utiliza la interfaz *VLC*, que facilita el trabajo con la biblioteca *LibVLC* para desarrollar el streaming de video y audio, así como para su almacenamiento.

Front Controller (Controlador Frontal): Todas las peticiones web son manejadas por un controlador frontal, que es el punto de entrada único para toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción definida en la clase *actions.class* y el nombre de un módulo con la URL entrado por el usuario, mientras otros componentes se encargan de la seguridad y la validación mediante el uso de los archivos *YML*. Se origina un flujo de eventos a partir del controlador frontal y se van delegando responsabilidades a cada componente hasta terminar en el *actions.class*. (41)

3.8 Diagrama de Despliegue.

El modelo de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, la comunicación entre ellos, y las instancias de los componentes y objetos que residen en estos nodos. Este modelo está constituido por:

- ✓ Nodos: Elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos.
- ✓ Dispositivos: Nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela.
- ✓ Conectores: Expresa el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo. (42)

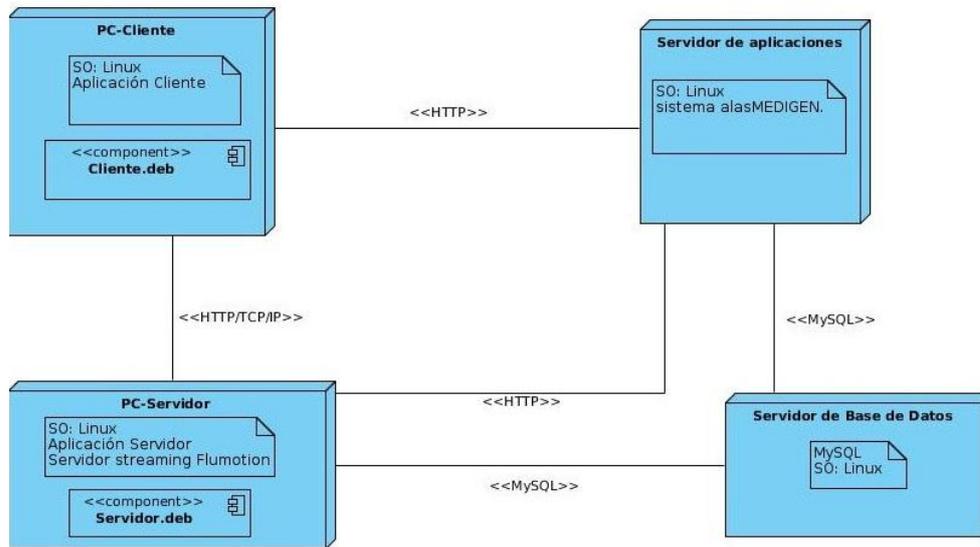


Fig. 13: Diagrama de despliegue.

En el diagrama de despliegue que se muestra en la Fig. 13 se representa la configuración de los nodos físicos sobre los cuales se despliega el componente audiovisual:

- ✓ El nodo PC-Cliente contiene la aplicación Cliente que se conecta con el nodo PC-Servidor donde está la aplicación Servidor a través del protocolo TCP/IP y se utiliza el protocolo HTTP para el envío de los flujos de video y audio.
- ✓ El nodo PC-Cliente se comunica con el nodo Servidor de aplicaciones donde se encuentra el sistema `alasMEDIGEN` a través del protocolo HTTP.
- ✓ El nodo Servidor de aplicaciones se comunica con el nodo PC-Servidor, donde se encuentra el servidor streaming `Flumotion` a través del protocolo HTTP.
- ✓ Al nodo Servidor de Base de Datos se conectan los nodos PC-Servidor y Servidor de aplicaciones a través del protocolo MySQL.

3.9 Conclusiones.

La modelación del diseño del sistema permitió describir la estructura de la implementación. Se desarrolló la vista lógica de la arquitectura, teniendo en cuenta las clases, los paquetes y los subsistemas del diseño. Se argumentaron y ejemplificaron los patrones de diseño utilizados que permitieron diseñar la solución del sistema. En el diagrama de despliegue se representaron los nodos y conexiones que intervienen en el funcionamiento del sistema, mostrando la configuración física sobre la que será desplegado el software. El diseño del sistema quedó conformado como base fundamental para la implementación.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.

En este capítulo se exponen las características de la implementación del componente a realizar. Se describe cómo los elementos del modelo de diseño se implementan en términos de componentes. Se definen los estilos de codificación a utilizar. Además se explican las pruebas a nivel de desarrollador realizadas al sistema.

4.1 Diagramas de Componentes.

Los diagramas de componentes modelan la vista estática de un sistema, describiendo sus elementos físicos y sus relaciones. Los componentes representan todos los tipos de software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, etc. Los elementos se agrupan en paquetes con vista a simplificar el modelado. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes, por lo que cada diagrama describe un apartado del sistema. (43)

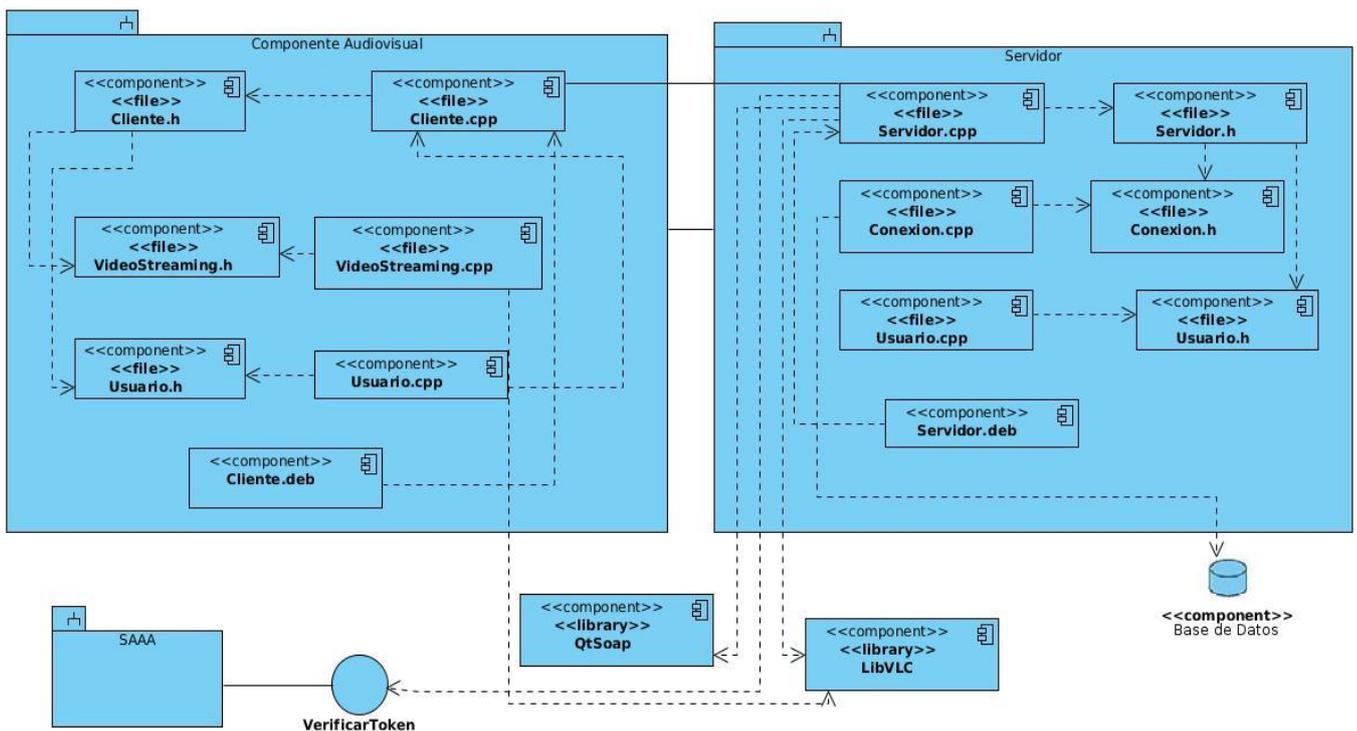


Fig. 14: Diagrama de componentes del CU Realizar videoconferencia.

El diagrama que se muestra en la Fig. 14 modela los componentes agrupados en paquetes, que interactúan en la implementación de la funcionalidad Realizar videoconferencia.

El paquete *Cliente* representa a la aplicación Cliente que contiene los siguientes componentes:

- ✓ *Cliente.h*: Representa la declaración de la clase *Cliente*.
- ✓ *Cliente.cpp*: Representa la implementación de la clase *Cliente*.
- ✓ Igualmente quedan definidos los componentes *VideoStreaming.h*, *VideoStreaming.cpp*, *Usuario.h*, *Usuario.cpp*.

El componente *QtSoap* y *LibVLC* representan a las bibliotecas utilizadas en la implementación del sistema.

El subsistema *Servidor*, que representa la aplicación Servidor de streaming. Está compuesto por los siguientes componentes:

- ✓ *Servidor.h*: Representa la declaración de la clase *Servidor*.
- ✓ *Servidor.cpp*: Representa la implementación de la clase *Servidor*.
- ✓ *Conexion.h*: Representa la declaración de la clase *Conexion*.
- ✓ *Conexion.cpp*: Representa la implementación de la clase *Conexion* para gestionar los datos persistentes.

Para consultar los restantes diagramas de componentes remitirse al expediente de proyecto, específicamente a la plantilla Arquitectura de software_v1.0.

4.2 Estándares de código.

Los estándares de codificación establecen un conjunto de reglas que los desarrolladores deben seguir para escribir el código fuente de un software. Su utilización posibilita que el sistema sea más legible y más fácil de mantener. Son pautas de programación que no están enfocadas a la lógica del programa, sino a la estructura y apariencia física, haciéndolo más reutilizable.

Para la codificación dentro de la implementación de los componentes, se utiliza el estándar de código del centro DATEC. A continuación se muestra el código de uno de los métodos de la clase *Servidor*, donde se visualizan algunas normas del estándar de código antes mencionado.

```
void Servidor::verificarPlanificacion() {
    if(conexion->obtenerIdSala(listaUsuario.at(listaUsuario.length()-1)->getUsuario()=="") {
        listaUsuario.removeAt(listaUsuario.length()-1);
        conexionCliente->close();
        conexionCliente->disconnectFromHost();
        return;
    }

    int idSala = conexion->obtenerIdSala(listaUsuario.at(listaUsuario.length()-1)->getUsuario());
    //Capturando la dirección ip del cliente
    QString ip= conexionCliente->peerAddress().toString();
    //Almacenando la dirección del cliente en la lista
    direccion.append(ip);

    // guardando la posición del solicitante.
    int posicion = listaUsuario.length()-1;
    bool esSolicitante= conexion->verificarSolicitante(listaUsuario.at(posicion)->getUsuario(), idSala);
    if (esSolicitante) {
        int c = listaUsuario.length();
        posicionSolicitante= c-1;
    }

    ui->listWidget->addItem(listaUsuario.at(listaUsuario.length()-1)->getNombre());
    ui->comenzarButton->setEnabled(true);
    conexionCliente->close();
    conexionCliente->disconnectFromHost();
}
```

Fig. 15: Ejemplo del estándar de código.

En este ejemplo se muestra la utilización de los espacios en el código para perfeccionar la organización del mismo. Además se escriben comentarios de una sola línea, los nombres de las variables comienzan con letras minúsculas y cada palabra relevante por la que esté compuesta se escribe con letra mayúscula. No existe un espacio entre el nombre de la función y el paréntesis izquierdo, ni entre este y la lista de parámetros. Se deja un espacio entre el paréntesis derecho y la llave del comienzo del cuerpo de la función. Las sentencias del cuerpo se escriben en la línea siguiente. La llave que cierra está alineada con la línea de declaración de la función y los nombres de las funciones se rigen por las mismas características que el de las variables.

4.3 Pruebas.

Las pruebas constituyen un factor importante en el proceso de desarrollo de un software. Estas permiten aumentar la calidad de los sistemas reduciendo el número de errores no detectados y el tiempo demorado en detectar los errores que puedan existir.

Para validar el componente audiovisual se realizaron las pruebas a nivel de desarrollador, ejecutando pruebas de unidad con el método de caja blanca. Además se desarrollaron pruebas a nivel de sistema, mediante el método de caja negra.

Prueba a nivel de desarrollador:

Es la prueba diseñada e implementada por el equipo de desarrollo. Tradicionalmente estas pruebas han sido consideradas solo para la prueba de unidad. Se aplica a los componentes del modelo de implementación para revisar el funcionamiento de todos los flujos de control y de datos. El diseño de casos de prueba de una unidad comienza una vez que se ha desarrollado, revisado y verificado su código. La prueba de unidad siempre está orientada a caja blanca.

Por su parte las pruebas de caja blanca se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa. Este método se centra en cómo diseñar los casos de prueba atendiendo al comportamiento interno y la estructura del programa.

La técnica utilizada en este método es la Prueba del camino básico, para ello es necesario conocer el número de caminos independientes de un determinado algoritmo mediante el cálculo de la complejidad ciclomática. (44)

A continuación se describe un ejemplo que muestra cómo se realiza este tipo de pruebas. Se analizó la clase *Servidor*, específicamente el procedimiento *verificarPlanificación()*. Este método comprueba si se puede permitir que un usuario determinado se conecte y participe en la videoconferencia. Es necesario enumerar las sentencias de código para construir un grafo asociado. Se le calcula la complejidad ciclomática mediante tres formas, para contar con un resultado seguro y confiable. (45)

$$V(G) = (A - N) + 2$$

$$V(G) = P + 13.$$

$$V(G) = R$$

Siendo "A" la cantidad total de aristas, "N" la cantidad total de nodos, "P" la cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas) y "R" la cantidad total de regiones.

```

void Servidor::verificarPlanificacion() {
1.0 if(conexion->obtenerIdSala(listaUsuario.at(listaUsuario.length()-1)->getUsuario()=="") {
2.0 listaUsuario.removeAt(listaUsuario.length()-1);
2.1 conexionCliente->close();
2.2 conexionCliente->disconnectFromHost();
2.3 return;
}
3.0 int idSala = conexion->obtenerIdSala(listaUsuario.at(listaUsuario.length()-1)->getUsuario());
3.1 //Capturando la dirección ip del cliente
3.1 QString ip= conexionCliente->peerAddress().toString();
3.2 //Almacenando la dirección del cliente en la lista
3.2 direccion.append(ip);
3.3 // guardando la posición del solicitante.
3.3 int posicion=listaUsuario.length()-1;
3.4 bool esSolicitante= conexion->verificarSolicitante(listaUsuario.at(posicion)->getUsuario(), idSala);
3.5 if (esSolicitante) {
4.0 int c = listaUsuario.length();
4.1 posicionSolicitante= c-1;
}
5.0 ui->listWidget->addItem(listaUsuario.at(listaUsuario.length()-1)->getNombre());
5.1 ui->comenzarButton->setEnabled(true);
5.2 conexionCliente->close();
5.3 conexionCliente->disconnectFromHost();
}6.0
    
```

Fig. 16: Ejemplo de prueba de caja blanca.

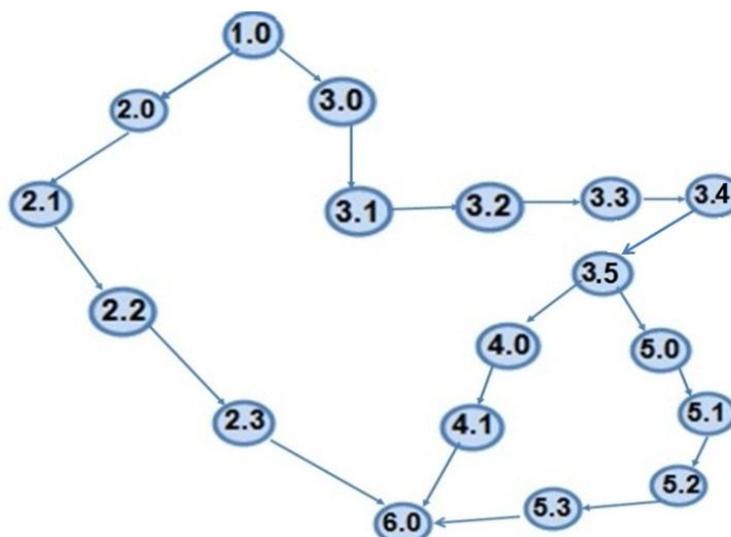


Fig. 17: Grafo del flujo de código para el caso de prueba: Verificar Planificación.

La cifra de la complejidad representa la cantidad de caminos por donde el flujo puede circular, estos deben de ser especificados y por último se ejecutan los casos de prueba para cada uno de ellos. En la función tomada como ejemplo en este caso existen tres caminos.

Caso de prueba para el primer camino básico:

Descripción: Se debe registrar el usuario, guardando su dirección IP y su posición en la lista de usuarios. Finalmente se debe desconectar al usuario.

Condición de ejecución: Se necesita buscar si el usuario que desea conectarse a la aplicación tiene una videoconferencia planificada para la fecha actual. Solo se guardará la posición en la lista de usuarios en caso de que tenga una planificación registrada en la base de datos y que sea el solicitante de la videoconferencia.

Entrada:

Usuario user=edit_mun_gm

Resultados esperados: Teniendo en cuenta que el usuario tiene una videoconferencia planificada para la fecha actual y es además el solicitante de la misma, se espera que se registre en la aplicación y se guarde la posición que ocupa en la lista de usuarios.

El usuario fue registrado y su posición fue guardada correctamente.

Para el segundo y tercer camino básico se mantienen los mismos datos en la descripción, en la condición de ejecución y en la entrada, cambiando los resultados esperados y la respuesta real del sistema.

Caso de prueba para el segundo camino básico:

Resultados esperados: Teniendo en cuenta que el usuario tiene una videoconferencia planificada para la fecha actual y no es el solicitante de la misma, se espera que se registre en la aplicación pero no se guarde la posición que ocupa en la lista de usuarios.

El usuario queda registrado en la aplicación, pero no se guardó su posición en la lista.

Caso de prueba para el tercer camino básico:

Resultados esperados: Teniendo en cuenta que el usuario no tiene una videoconferencia planificada para la fecha actual se espera que no se registre en la aplicación.

El usuario no se registra en la aplicación.

Luego de aplicar los distintos casos de pruebas, se pudo comprobar que el flujo de trabajo de la función está correcto ya que cumple con las condiciones necesarias que se habían planteado. Es importante mencionar que para realizar estas pruebas se realiza un proceso de depuración del código utilizando puntos de ruptura.

Prueba a nivel de sistema:

Son las pruebas que se hacen cuando se obtiene el programa final, para verificar cada una de sus funcionalidades. En este nivel de prueba el software debe integrarse con el hardware correspondiente. La validación del software orientado a objeto se centra en las acciones visibles del usuario y las salidas del sistema reconocibles por este. Para realizar los casos de prueba de sistema, el ejecutor debe basarse en los casos de uso que forman parte del modelo de análisis. Esto permite verificar que el sistema en desarrollo cumpla o no con sus especificaciones. Los métodos convencionales de prueba de caja negra, son los más usados para dirigir estas pruebas. Las pruebas de caja negra se centran en los requisitos funcionales del sistema e ignoran la estructura de control. Pretenden demostrar que: (45)

- ✓ Las funciones del software son operativas.
- ✓ La entrada se acepta de forma adecuada.
- ✓ Se produce una salida correcta.
- ✓ La integridad de la información externa se mantiene.

La técnica utilizada en este método es Partición de equivalencia, que divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.

Se desarrollan casos de prueba para este método según las funcionalidades de la aplicación. Partiendo de la descripción de los casos de uso del sistema, como apoyo para las revisiones, se diseña un caso de prueba asociado a cada caso de uso. Para detallar un caso de prueba se utiliza una tabla para facilitar la ejecución del mismo. (44)

A continuación se muestra la tabla correspondiente al caso de prueba del caso de uso: Autenticar usuario.

Escenario	Descripción	Usuario	Contraseña	Respuesta del sistema	Flujo central
EC Introducir datos usuario y contraseña	1.1 Verificar en el RIS el usuario y la contraseña introducidos.	V medigen	V 1234567	El sistema muestra la interfaz con el formulario para conectarse al servidor de	1. Introducir los datos del usuario y la contraseña. 2. Seleccionar el botón "Iniciar

				streaming.	Sesión”.
EC 1.2 Dejar datos en blanco e insertar datos incorrectos	Dejar datos en blanco e insertar datos incorrectos	I	V	El sistema muestra el mensaje: “Usuario o contraseña no válidos” y limpia los campos del usuario y la contraseña.	1. Introducir los datos del usuario y la contraseña. 2. Seleccionar el botón “Iniciar Sesión”.
		medi	1234567		
		V	I		
		medigen	123		

Tabla 8: Caso de prueba del caso de uso: Autenticar usuario.

En la siguiente tabla se describen las variables utilizadas en el caso de prueba anterior:

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Usuario	campo de texto	No	El usuario debe pertenecer al RIS, para poder ser autenticado en el sistema.
2	Contraseña	campo de texto	No	Debe de ocultar los caracteres introducidos en formato de contraseña y debe de coincidir con la contraseña del usuario en el RIS.

Tabla 9: Descripción de las variables del caso de prueba: Autenticar usuario.

Para consultar los restantes casos de prueba remitirse al expediente de proyecto.

Luego de realizadas las pruebas al sistema se detectaron algunas no conformidades (NC), las cuales se muestran a continuación:



Capítulo 4: Implementación y Prueba

Elemento	No.	No Conformidad	Aspecto correspondiente.	Etapa de detección	Clasificación.	Estado NC.
Aplicación	1	Si el usuario no introduce los datos correctos para registrarse en el sistema, se debe mostrar las excepciones propias del RIS	Autenticar usuario	Pruebas de funcionalidad y confiabilidad	Significativa	PD: 18 de marzo de 2012 RA: 20 de marzo de 2012
Aplicación	2	En la sección Buscar videoconferencia del sistema alasMEDIGEN, cuando se muestren las videoconferencias se debe visualizar la fecha y la hora de las mismas.	Mostrar videoconferencia bajo demanda	Pruebas de funcionalidad y confiabilidad	Significativa	PD: 18 de marzo de 2012 RA: 23 de marzo de 2012
Aplicación	3	Cuando se muestre la videoconferencia la imagen del solicitante debe salir en el fondo de la misma.	Realizar videoconferencia	Pruebas de funcionalidad y confiabilidad	No Significativa	PD: 18 de marzo de 2012 RA: 23 de marzo de 2012
Aplicación	4	Cuando se muestre la videoconferencia debe mostrarse el nombre del solicitante en el fondo	Realizar videoconferencia	Pruebas de funcionalidad y confiabilidad	No Significativa	PD: 18 de marzo de 2012 RA: 27 de marzo de 2012

Aplicación	5	En los mensajes deben mostrarse las tildes y los caracteres extraños que se necesiten.	Realizar videoconferencia	Pruebas de funcionalidad y confiabilidad	No Significativa	PD: 27 de marzo de 2012 RA: 28 de marzo de 2012
✓ <u>PD</u> : Prueba pendiente. ✓ <u>RA</u> : Prueba resuelta.						

Tabla 10: Descripción de las no conformidades de la aplicación.

4.4 Conclusiones.

En este capítulo se obtuvieron los diagramas de componentes, representándose los componentes físicos de la aplicación y las relaciones entre estos. Además se mostró el estándar de código utilizado, el cual contribuyó a la obtención de un código fácil de comprender. Se realizaron las pruebas al sistema a nivel de desarrollador a través de los métodos de caja blanca y caja negra, que permitieron detectar los errores que fueron corregidos posteriormente, así como comprobar la calidad de la implementación realizada.

CONCLUSIONES

El estudio de los sistemas existentes de Teleconsulta, facilitó la selección de las herramientas y las tecnologías a utilizar para el desarrollo del componente audiovisual.

El análisis de las funcionalidades del sistema y la utilización de los diferentes patrones de diseño y arquitectura permitieron diseñar una solución considerando las buenas prácticas de desarrollo de software.

El producto funcional obtenido a partir de la implementación de las clases del diseño permitió desarrollar un componente audiovisual integrado a alasMEDIGEN.

Se realizaron las pruebas a nivel de desarrollador que permitieron identificar y corregir los errores del sistema, así como verificar la correcta implementación de las funcionalidades del mismo.

El componente audiovisual permite a los especialistas tener un contacto visual y acústico con los pacientes, permitiéndoles realizar los diagnósticos de manera más precisa.



RECOMENDACIONES

Al término de esta investigación se recomienda:

- ✓ Crear un servicio en alasMEDIGEN que permita verificar si un usuario pertenece a una Teleconsulta planificada, para que pueda ser utilizado en el componente audiovisual.

REFERENCIAS BIBLIOGRAFICAS

1. Lexy Gaspar Cárdenas, Ezequiel González Pupo. Propuesta de implementación de nuevos medios de comunicación para la Teleconsulta genética de alasMEDIGEN. 2009.
2. Telemedicina: Introducción, aplicación y principios de desarrollo. Ruiz Ibáñez, Carlos, Zuluaga de Cadena, Ángela y Trujillo Zea, Andrés. 1, 2007, Vol. 21.
3. Telemedicine: Technology and Clinical Applications. Perednia, D y Allen, A. 273:483, JAMA : s.n., Vol. 8.
4. Programa para Videoconferencia. [En línea] [Citado el: 5 de 11 de 2011.] <http://videoconferenciagvo.com/>.
5. Yudiel La Rosa González, Yusdenis Sánchez Perodín, Elvismary Molina De Armas, Dayana Joseph. alasMEDIGEN 1.2: "Sistema Informático de Genética Médica". s.l. : Universidad de las Ciencias Informáticas (UCI).
6. Teleconsulta de lesiones pigmentadas. Ramírez Moreno, David. 9, Piel, 2004, Vol. 19.
7. TESIS. Sistema de Teleconsulta. [En línea] [Citado el: 10 de 11 de 2011.] <http://www.tesis.es/index.php?seccion=2&subseccion=6>.
8. Conferencia Ministerial sobre la Sociedad de la Información de América Latina y el Caribe. División de Desarrollo Productivo y Empresarial de la Comisión Económica para América Latina y el Caribe (CEPAL). Cimoli, Mario. Lima : s.n., 2010.
9. Comitas Comunicaciones S.A. [En línea] [Citado el: 8 de 11 de 2011.] <http://www.comitas.es/Menu.aspx?Menu=Principal>.
10. ANDROME.ES. Phemium Consultant. [En línea] [Citado el: 10 de 11 de 2011.] http://www.androme.es/cas/prod_phem_consultant.htm.
11. GVO Conference. [En línea] [Citado el: 10 de 11 de 2011.] <http://jfschiller.gvoconference.com/es/features.php>.
12. Axis Communications. Software de Gestión de videos. [En línea] [Citado el: 11 de 11 de 2011.] <http://www.axis.com/es/products/video/software/index.htm>.
13. Infomed. [En línea] [Citado el: 11 de 11 de 2011.] <http://www.sld.cu/telemedicina/indice.html>.
14. Gaspar Cárdenas, Lexy y González Pupo, Ezequiel. Propuesta de implementación de nuevos medios de comunicación para la Teleconsulta genética de alasMEDIGEN. 2009.
15. Andrés López, José Luis y Fuster Badia, Enrique. Sistemas Multimedias. Streaming Server (Streaming de video y audio).
16. Matamoro Hechavarría, Isidro de Jesús. Arquitectura del Sistema de Videoconferencia del Departamento de Señales Digitales. 2011.

17. Flumotion. [En línea] [Citado el: 06 de 05 de 2012.] <http://www.flumotion.net/>.
18. Parte I. El Proceso Unificado de Modelado. [aut. libro] Ivar Jacobson, James Rumbaugh y Gradys Booch. El Proceso Unificado de Modelado.
19. Lenguaje Unificado de Modelado. [aut. libro] Ivar Jacobson, James Rumbaugh y Gradys Booch.
20. Cplusplus. [En línea] [Citado el: 25 de 04 de 2012.] www.cplusplus.com/info/description.
21. My PHP.net. [En línea] [Citado el: 2012 de 02 de 17.] <http://www.php.net/manual/es/faq.general.php>.
22. NetBeans. [En línea] [Citado el: 12 de 11 de 2011.] http://netbeans.org/community/releases/61/index_es.html.
23. Zona qt. [En línea] [Citado el: 16 de 04 de 2012.] <http://www.zonaqt.com>.
24. POTENCIER, F. y ZANINOTTO, F. Symfony, la guía definitiva. 2007.
25. Visual Paradigm. [En línea] [Citado el: 25 de 11 de 2011.] www.visual-paradigm.com/product/vpuml.
26. QT. [En línea] [Citado el: 04 de 05 de 2012.] qt.nokia.com/products/developer-tools.
27. NetBeans.org. [En línea] [Citado el: 02 de 04 de 2012.] <http://netbeans.org/features/ide/editor.html>.
28. Oracle. [En línea] [Citado el: 26 de 11 de 2011.] www.oracle.com/us/products/mysql/index.html.
29. Kabir, M. La Biblia del Servidor Apache 2.0. 2003.
30. Dirección nacional de servicios académicos virtuales. Universidad Nacional de Colombia. [En línea] [Citado el: 2012 de 02 de 17.] <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060029/lecciones/cap6-3.html>.
31. Jiménez López, Rafael. Análisis y diseño de un framework para el modelado estadístico. 2003.
32. QT Developer Network. [En línea] [Citado el: 04 de 05 de 2012.] <http://qt-project.org/doc/qt-4.8/model-view-programming.html>.
33. Entorno Virtual de Desarrollo . [En línea] [Citado el: 2011 de 11 de 12.] http://eva.uci.cu/file.php/161/Documentos/Materiales_basicos/Materiales_basicos_de_la_Unidad_2/Swebok_ESPANOL_Capitulo_2_Requerimientos_de_Software.pdf.
34. Profesor Lauro Soto, Ensenada, BC, Mexico. Tecnológico. [En línea] <http://www.mitecnologico.com/Main/ProtocoloHttp>.
35. Biblioteca virtual de derecho economía y ciencias sociales. Universidad de Málaga. [En línea] [Citado el: 2012 de 02 de 15.]

- <http://www.eumed.net/libros/2009c/584/RUP%20Diseno%20e%20implementacion%20del%20sistema.htm>.
36. Arquitectura de redes, Laboratorio y Departamento de Automática, Universidad de Alcalá. Introducción de Sockets en C.
37. Kruchten, Philippe. Planos Arquitectónicos: El Modelo de “4+1” Vistas de la Arquitectura del Software. [En línea] [Citado el: 15 de 04 de 2012.] http://www.oscargallardo.com/wp-content/uploads/2011/04/Modelo4_1Krutchen.pdf.
38. Recasens, Lluís Martínez. Diseño e Implementación de Módulos para un Sistema de TV P2P. s.l. : Universidad Politécnica de Catalunya, 2008.
39. Mairelys Fernández González, Osley Zorrilla Rivera. Diseño e implementación del componente Ajuste al Costo del Subsistema Costos y Procesos del Sistema Integral de Gestión de Entidades CEDRUX. Habana : UCI, 2010.
40. Fabien Potencier, François Zaninotto. Symfony, la guía definitiva. 2008.
41. Mühlrad, Daniel. Patrones de diseño. 25 de Mayo 2008.
42. Oquendo, Yisel Ybarra Cristiá Eddy Eliceo Alvarado. Desarrollo de un componente de Transmisión de Audio y Video para el Sistema de Teleconsulta. La Habana : s.n., junio 2011.
43. Universidad de castilla-La mancha. [En línea] Departamento de Sistemas Informáticos. [Citado el: 16 de 04 de 2012.] <http://www.dsi.uclm.es/>.
44. Javier J. Gutiérrez, María J. Escalona, Manuel Mejías, Arturo H. Torres, Jesús Torres. Implementación de pruebas del sistema. s.l. : Universidad de Sevilla., 2007.
45. Lamancha, Beatriz Pérez. Actas de Talleres de Ingeniería del Software y Bases de Datos. s.l. : Centro de Ensayos de Software. Universidad de la República, Montevideo, Uruguay, 2007. Vol. 1, No. 4.
46. Empresa de software Zator Systems. [En línea] [Citado el: 25 de 03 de 2012.] http://www.zator.com/Internet/N_11.htm.
47. Desarrollo Web. [En línea] [Citado el: 04 de 05 de 2012.] <http://www.desarrolloweb.com/articulos/que-es-html5.html>.
48. Sande, Martín. Programación en C++ con Qt bajo Entorno GNU/Linux. 2004.
49. Trolltech. Qt 4.0 Whitepaper. 2005.
50. Jasmin Blanchette, Mark Summerfield, Trolltech Press. C++ GUI Programming with Qt 4. s.l. : Trolltech AS, 2006.
51. Sinn., Richard. Real Time Streaming Protocol. 2006.

52. Meggelen., Jim Van. Building a Simple, Cost-Effective Conferencing Server Using Sangoma Wanpipe and YATE (Yet Another Telephony Engine). s.l. : Core Telecom Innovations Inc., September 18, 2008.
53. Murray Cumming, Thomas Vander Stichele, and Christian Fredrik Kalager Schaller. Flumotion manual. 2008.
54. QT home, QtSoapMessage Class Reference. s.l. : Nokia, 2009.
55. VideoLan, Documentation: Modules/v4l, . 2009.
56. Aggio, Santiago. Implementación de una MCU por software en ARIU. . s.l. : Jornadas Técnicas de ARIU., Mayo de 2010.
57. V., Felipe López Placencio. Agustín González. “Servicio automatizado para streaming directo y bajo demanda”, Universidad . s.l. : Técnica Federico Santa María. Departamento de electrónica. , 23/05/2011.
58. Webcam, community . s.l. : Ubuntu documentation, 2012.
59. Red5 Media Server. [En línea] [Citado el: 15 de 01 de 2012.] www.red5.org.
60. Recasens, Lluís Martínez. Diseno e Implementación de Módulos para un Sistema de TV P2P . s.l. : Universidad Politécnic de Catalunya, 2008.
61. Bartolomé, María del Carmen, y otros, y otros. Desarrollo de Servicios Avanzados de Voz sobre redes de Paquetes. Telemática. Departamento de Ingeniería. s.l. : Universidad Carlos III de Madrid., 2006.

BIBLIOGRAFÍA

1. Lexy Gaspar Cárdenas, Ezequiel González Pupo. Propuesta de implementación de nuevos medios de comunicación para la Teleconsulta genética de alasMEDIGEN. 2009.
2. Telemedicina: Introducción, aplicación y principios de desarrollo. Ruiz Ibáñez, Carlos, Zuluaga de Cadena, Ángela y Trujillo Zea, Andrés. 1, 2007, Vol. 21.
3. Telemedicine: Technology and Clinical Applications. Perednia, D y Allen, A. 273:483, JAMA : s.n., Vol. 8.
4. Programa para Videoconferencia. [En línea] [Citado el: 5 de 11 de 2011.] <http://videoconferenciagvo.com/>.
5. Yudiel La Rosa González, Yusdenis Sánchez Perodín, Elvismary Molina De Armas, Dayana Joseph. alasMEDIGEN 1.2: "Sistema Informático de Genética Médica". s.l. : Universidad de las Ciencias Informáticas (UCI).
6. Teleconsulta de lesiones pigmentadas. Ramírez Moreno, David. 9, Piel, 2004, Vol. 19.
7. TESIS. Sistema de Teleconsulta. [En línea] [Citado el: 10 de 11 de 2011.] <http://www.tesis.es/index.php?seccion=2&subseccion=6>.
8. Conferencia Ministerial sobre la Sociedad de la Información de América Latina y el Caribe. División de Desarrollo Productivo y Empresarial de la Comisión Económica para América Latina y el Caribe (CEPAL). Cimoli, Mario. Lima : s.n., 2010.
9. Comitas Comunicaciones S.A. [En línea] [Citado el: 8 de 11 de 2011.] <http://www.comitas.es/Menu.aspx?Menu=Principal>.
10. ANDROME.ES. Phemium Consultant. [En línea] [Citado el: 10 de 11 de 2011.] http://www.androme.es/cas/prod_phem_consultant.htm.
11. GVO Conference. [En línea] [Citado el: 10 de 11 de 2011.] <http://jfschiller.gvoconference.com/es/features.php>.
12. Axis Communications. Software de Gestión de videos. [En línea] [Citado el: 11 de 11 de 2011.] <http://www.axis.com/es/products/video/software/index.htm>.
13. Infomed. [En línea] [Citado el: 11 de 11 de 2011.] <http://www.sld.cu/telemedicina/indice.html>.
14. Gaspar Cárdenas, Lexy y González Pupo, Ezequiel. Propuesta de implementación de nuevos medios de comunicación para la Teleconsulta genética de alasMEDIGEN. 2009.
15. Andrés López, José Luis y Fuster Badia, Enrique. Sistemas Multimedias. Streaming Server (Streaming de video y audio).
16. Matamoro Hechavarría, Isidro de Jesús. Arquitectura del Sistema de Videoconferencia del Departamento de Señales Digitales. 2011.

17. Flumotion. [En línea] [Citado el: 06 de 05 de 2012.] <http://www.flumotion.net/>.
18. Parte I. El Proceso Unificado de Modelado. [aut. libro] Ivar Jacobson, James Rumbaugh y Gradys Booch. El Proceso Unificado de Modelado.
19. Lenguaje Unificado de Modelado. [aut. libro] Ivar Jacobson, James Rumbaugh y Gradys Booch.
20. Cplusplus. [En línea] [Citado el: 25 de 04 de 2012.] www.cplusplus.com/info/description.
21. My PHP.net. [En línea] [Citado el: 2012 de 02 de 17.] <http://www.php.net/manual/es/faq.general.php>.
22. NetBeans. [En línea] [Citado el: 12 de 11 de 2011.] http://netbeans.org/community/releases/61/index_es.html.
23. Zona qt. [En línea] [Citado el: 16 de 04 de 2012.] <http://www.zonaqt.com>.
24. POTENCIER, F. y ZANINOTTO, F. Symfony, la guía definitiva. 2007.
25. Visual Paradigm. [En línea] [Citado el: 25 de 11 de 2011.] www.visual-paradigm.com/product/vpuml.
26. QT. [En línea] [Citado el: 04 de 05 de 2012.] qt.nokia.com/products/developer-tools.
27. NetBeans.org. [En línea] [Citado el: 02 de 04 de 2012.] <http://netbeans.org/features/ide/editor.html>.
28. Oracle. [En línea] [Citado el: 26 de 11 de 2011.] www.oracle.com/us/products/mysql/index.html.
29. Kabir, M. La Biblia del Servidor Apache 2.0. 2003.
30. Dirección nacional de servicios académicos virtuales. Universidad Nacional de Colombia. [En línea] [Citado el: 2012 de 02 de 17.] <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060029/lecciones/cap6-3.html>.
31. Jiménez López, Rafael. Análisis y diseño de un framework para el modelado estadístico. 2003.
32. QT Developer Network. [En línea] [Citado el: 04 de 05 de 2012.] <http://qt-project.org/doc/qt-4.8/model-view-programming.html>.
33. Entorno Virtual de Desarrollo . [En línea] [Citado el: 2011 de 11 de 12.] http://eva.uci.cu/file.php/161/Documentos/Materiales_basicos/Materiales_basicos_de_la_Unidad_2/Sw_ebok_ESPANOL_Capitulo_2_Requerimientos_de_Software.pdf.
34. Profesor Lauro Soto, Ensenada, BC, Mexico. Tecnológico. [En línea] <http://www.mitecnologico.com/Main/ProtocoloHttp>.
35. Biblioteca virtual de derecho economía y ciencias sociales. Universidad de Málaga. [En línea] [Citado el: 2012 de 02 de 15.]

- <http://www.eumed.net/libros/2009c/584/RUP%20Diseno%20e%20implementacion%20del%20sistema.htm>.
36. Arquitectura de redes, Laboratorio y Departamento de Automática, Universidad de Alcalá. Introducción de Sockets en C.
37. Kruchten, Philippe. Planos Arquitectónicos: El Modelo de “4+1” Vistas de la Arquitectura del Software. [En línea] [Citado el: 15 de 04 de 2012.] http://www.oscargallardo.com/wp-content/uploads/2011/04/Modelo4_1Krutchen.pdf.
38. Recasens, Lluís Martínez. Diseño e Implementación de Módulos para un Sistema de TV P2P. s.l. : Universidad Politécnica de Catalunya, 2008.
39. Mairelys Fernández González, Osley Zorrilla Rivera. Diseño e implementación del componente Ajuste al Costo del Subsistema Costos y Procesos del Sistema Integral de Gestión de Entidades CEDRUX. Habana : UCI, 2010.
40. Fabien Potencier, François Zaninotto. Symfony, la guía definitiva. 2008.
41. Mühlrad, Daniel. Patrones de diseño. 25 de Mayo 2008.
42. Oquendo, Yisel Ybarra Cristiá Eddy Eliceo Alvarado. Desarrollo de un componente de Transmisión de Audio y Video para el Sistema de Teleconsulta. La Habana : s.n., junio 2011.
43. Universidad de castilla-La mancha. [En línea] Departamento de Sistemas Informáticos. [Citado el: 16 de 04 de 2012.] <http://www.dsi.uclm.es/>.
44. Javier J. Gutiérrez, María J. Escalona, Manuel Mejías, Arturo H. Torres, Jesús Torres. Implementación de pruebas del sistema. s.l. : Universidad de Sevilla., 2007.
45. Lamancha, Beatriz Pérez. Actas de Talleres de Ingeniería del Software y Bases de Datos. s.l. : Centro de Ensayos de Software. Universidad de la República, Montevideo, Uruguay, 2007. Vol. 1, No. 4.
46. Empresa de software Zator Systems. [En línea] [Citado el: 25 de 03 de 2012.] http://www.zator.com/Internet/N_11.htm.
47. Desarrollo Web. [En línea] [Citado el: 04 de 05 de 2012.] <http://www.desarrolloweb.com/articulos/que-es-html5.html>.
48. Sande, Martín. Programación en C++ con Qt bajo Entorno GNU/Linux. 2004.
49. Trolltech. Qt 4.0 Whitepaper. 2005.
50. Jasmin Blanchette, Mark Summerfield, Trolltech Press. C++ GUI Programming with Qt 4. s.l. : Trolltech AS, 2006.
51. Sinn., Richard. Real Time Streaming Protocol. 2006.

52. Meggelen., Jim Van. Building a Simple, Cost-Effective Conferencing Server Using Sangoma Wanpipe and YATE (Yet Another Telephony Engine). s.l. : Core Telecom Innovations Inc., September 18, 2008.
53. Murray Cumming, Thomas Vander Stichele, and Christian Fredrik Kalager Schaller. Flumotion manual. 2008.
54. QT home, QtSoapMessage Class Reference. s.l. : Nokia, 2009.
55. VideoLan, Documentation: Modules/v4l, . 2009.
56. Aggio, Santiago. Implementación de una MCU por software en ARIU. . s.l. : Jornadas Técnicas de ARIU., Mayo de 2010.
57. V., Felipe López Placencio. Agustín González. “Servicio automatizado para streaming directo y bajo demanda”, Universidad . s.l. : Técnica Federico Santa María. Departamento de electrónica. , 23/05/2011.
58. Webcam, community . s.l. : Ubuntu documentation, 2012.
59. Red5 Media Server. [En línea] [Citado el: 15 de 01 de 2012.] www.red5.org.
60. Recasens, Lluís Martínez. Diseno e Implementación de Módulos para un Sistema de TV P2P . s.l. : Universidad Politécnic de Catalunya, 2008.
61. Bartolomé, María del Carmen, y otros, y otros. Desarrollo de Servicios Avanzados de Voz sobre redes de Paquetes. Telemática. Departamento de Ingeniería. s.l. : Universidad Carlos III de Madrid., 2006.



GLOSARIO DE TÉRMINOS

API: Interfaz de Programación de Aplicaciones.

Auscultación: Procedimiento clínico de la exploración física que consiste en escuchar de manera directa el área torácica o el abdomen en busca de diferentes sonidos patológicos o normales.

Difusión: Forma de transmisión de información, donde un nodo emisor envía información a una multitud de nodos receptores de manera simultánea.

FTP: Protocolo de transferencia de archivos, permite transmitir archivos entre sistemas conectados a una red y sus siglas vienen dadas por su nombre en inglés File Transfer Protocol.

GStreamer: Framework multimedia que permite crear aplicaciones audiovisuales.

GUI: Interfaz gráfica de usuario que permite a través del uso y la representación visual, la interacción de forma amigable entre el usuario y el sistema informático, sus siglas vienen dadas por su nombre en inglés Graphical User interface.

Infomed: Red Telemática de Salud en Cuba.

Mosaico: Obra realizada con fracciones diversas, en este caso, se crea el mosaico con los flujos de video y audio de cada usuario.

Red5: Un servidor de código abierto para entregar contenido streaming en flash.

RIS: Es un sistema que gestiona la información centralizada de salud.

RTP: Protocolo de transporte de tiempo real, sus siglas vienen dadas por su nombre en inglés de Real Time Transport Protocol.

RTSP: Protocolo de flujo en tiempo real, establece y controla uno o muchos flujos sincronizados, sus siglas vienen dadas por su nombre en inglés Real Time Streaming Protocol.

SOAP: Protocolo de comunicación entre aplicaciones a través de mensajes vía Internet cuyas siglas vienen dadas por su nombre en inglés Simple Object Access Protocol.

Socket: nombre de una API para la familia de protocolos TCP/IP que permite que dos programas pueden intercambiar cualquier flujo de datos.

Softel: Empresa cubana que ofrece soluciones informáticas para el Sistema de Salud.

UDP: Protocolo de datagrama de usuario, que pertenece al nivel de transporte basado en el intercambio de datagramas, sus siglas vienen dadas por su nombre en inglés User Datagram Protocol.