

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 6



Título: Multiplexor de flujos de video provenientes de cámaras IP

**Trabajo de Diploma para optar por el Título de Ingeniero en
Ciencias Informáticas**

Autores: Reidel Morales Toledo

Serguey Caballero Sarduy

Tutor(es): Ing. Reynier Pupo Gómez

Msc. Rafael Leodán Cardero Álvarez

La Habana, 2015

“Año 57 de la Revolución”

“Considero más valiente al que conquista sus deseos que al que conquista a sus enemigos, ya que la victoria más dura es la victoria sobre uno mismo”

Aristóteles

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste se firma la presente a los ____ días del mes de _____ del año _____.

Autor: Reidel Morales Toledo

Autor: Serguey Caballero Sarduy

Tutor: Ing. Reynier Pupo Gómez

Tutor: Msc. Rafael Leodán Cardero Álvarez

Datos de Contacto

Tutor: Ing. Reynier Pupo Gómez

Ingeniero en Ciencias Informáticas, alcanzó el Nivel Superior en el año 2010 en la Universidad de las Ciencias Informáticas, La Habana, Cuba. Cuenta con 5 años de experiencia en el desarrollo de aplicaciones para el procesamiento de videos y sistemas de video vigilancia dentro del Centro Geoinformática y Señales Digitales (GEYSED) de la Universidad de las Ciencias Informáticas. Se desempeña actualmente como Jefe del Proyecto Xilema Suria.

Correo electrónico: rgomez@uci.cu

Co-Tutor: Msc. Rafael Leodán Cardero Álvarez

Ingeniero en Ciencias Informáticas (2008). Máster en Informática Aplicada (2011). Trabaja en lo referido al procesamiento de imágenes digitales y el reconocimiento de patrones en general. Ha participado en varios eventos afines a su campo de interés.

Correo electrónico: rlcordero@uci.cu

Dedicatoria

A mis padres que siempre han confiado en mí y me han dado todo su apoyo, a los cuales hoy les debo lo que soy, y a mis hermanos a los cuales amo por siempre estar ahí cuando los necesito y por ser una parte tan importante de mí.

Reidel

A mi Mamá por su amor, su dedicación y sacrificio, por apoyarme en todo momento, por haber estado siempre a mi lado en cada problema.

A mi Papá por ayudarme a levantarme luego de cada caída y por enseñarme a ser una mejor persona.

Serguey

Agradecimientos

A mi mamá y a mi papá, por ser los mejores padres del mundo, por darme todo su amor, su cariño y apoyo incondicional, por todo lo que se esforzaron y me ayudaron en estos 5 años para que hoy me pueda estar graduando de Ingeniero Informático. Por siempre darme fuerzas para seguir en los momentos difíciles, por sus consejos, por su preocupación, por estar presente en todos los momentos de mi vida, por luchar por mí y preocuparse por mi futuro y por luchar siempre por darme lo mejor.

A mis hermanos Odelay y Reinel, y hasta a mi sobrinita Cynthia, quienes forman una parte muy importante de mí, por quererme y respetarme y estar siempre presente cuando los he necesitado.

A mi novia Doina por estar conmigo en los buenos y malos momentos, escuchándome, ayudándome, por quererme y apoyarme desde el inicio y ser además mi mejor amiga.

A mis suegros, Leonardo, a Anita y Alejandro por haberme acogido con tanto cariño y motivarme semana tras semana a seguir adelante y dar lo mejor de mí. Han sido otra familia para mí.

A mi compañero de tesis, Serguey, por ayudarme a ser un mejor informático, por ser constante en el trabajo conmigo, y sobre todo por ser el mejor compañero de tesis que pude tener. Por su hermandad incondicional que siempre perdurará.

A nuestros tutores Pupo, Rafael y también al Yorbe, por su ayuda y preocupación, por estar siempre a nuestra disposición, prácticamente han sido un tesista más, que han trabajado a la par con nosotros.

A mis amigos que hoy no están presentes en la universidad pero fueron excelentes compañeros, especialmente al Yuli, al Rafa y a Mike, el que por lejos que esté, siempre va a ser mi hermano y nunca olvidaré estos años de una amistad que nunca terminará.

A los muchos hermanos que he hecho en la inolvidable UCI, a los que me acompañan desde que llegue aquí, Daldis, Rolo, Frank, Roger, Eric, Bidot, Trujillo, Eduardo, Alain, Sonia, Alex y a los que conocí después, Serguey, Ronald, Noslen, Jorjito, Pedro, Adonis, Yasiel, Yoan, al piquete del futbol y nuestras interminables discusiones que espero que continúen aunque sea por Facebook, Raciél, Karel, y Randy, la famosa RKR.

A todo el equipo de trabajo del proyecto Video Vigilancia, con quienes tuve el placer de trabajar y que ayudaron de una forma u otra con la realización de esta tesis.

Agradecimientos

A nuestra oponente Janet Cristina, que nos supo corregir y ayudar cuando lo requerimos y ayudó a que hoy podamos estar aquí.

A los profesores del tribunal que con sus críticas y consejos nos ayudaron a realizar esta tesis.

A todas mis amistades del año, a las buenas amistades de la 7ma graduación de la UCI y a todos aquellos que no menciono, pero que han formado parte de mi estancia en esta universidad y a quienes agradezco cada minuto compartido.

Gracias.

Reidel

Agradecimientos

A mi mamá por ser la mujer más especial del mundo, por todo el sacrificio que ha hecho para que yo pueda estar aquí, por ser la que me da fuerza y porque es lo más importante de mi vida, te quiero mamá.

A mi papá por depositar toda su confianza en mí, por su preocupación, por tratar siempre de darme lo mejor y por enseñarme muchas cosas de la vida.

Al Nene por considerarme como un hijo, ayudarme incondicionalmente como un padre y estar levantándome el ánimo siempre.

A mis hermanos Yasmany y Sudy por apoyarme y exhortarme a seguir adelante.

A mis abuelos por toda la sabiduría y conocimientos que me han aportado.

A Mamita que me animaba diciéndome que me faltaba poco, cuando todavía estaba en primero.

A mis tías por preocuparse por mí en el transcurso de estos cinco años.

A mi primo Yordanky al que quiero y aprecio como un hermano.

A mi hermano del alma, Ronald, porque no hubiese aprendido ni la mitad de lo que sé sino fuera por él, por hacer de su casa otra casa para mí, por soportarme 5 años y hacerme sentir como uno más de su familia.

A mi otro hermano Alexito por ayudarme siempre que lo molesté, por la compañía en cada uno de los viajes y por las miles de horas que hemos compartido desde primer año.

A mi compañero de tesis, Reidel, por levantarme el ánimo luego de cada corte de tesis y demostrarme que 255 comentarios no son nada.

A Mike por enseñarme a apreciar el séptimo arte, porque siempre estuvo ahí para conversar cuando más abatido estaba y por recomendarme Friends.

A nuestros tutores Pupo y Rafael por su ayuda y preocupación, por estar siempre dispuestos, porque prácticamente han sido un tesista más y han trabajado a la par con nosotros.

A Yorbe que lo único que le faltó para ser otro tutor fue incluir su nombre en el documento de tesis.

Agradecimientos

A mis amigos Rolando, Noslen, Elibetsy, Yasiel, Alejandro, Abel, Adonis, Asiel y a todos los que no menciono por problemas de espacio o de memoria.

A todos aquellos que no menciono, pero que han formado parte de mi estancia en esta universidad y a quienes agradezco cada minuto compartido.

Gracias.

Serguey

Resumen

El producto Xilema Suria es un sistema de video vigilancia soportado sobre tecnología IP¹. Su base está formada por estaciones de monitoreo simultáneo en tiempo real de los diferentes flujos de video generados por cada una de las cámaras de seguridad. El desarrollo de un multiplexor de video IP para el sistema Xilema Suria tiene como objetivo lograr una aplicación que multiplexe los flujos de video transmitidos por las cámaras IP. El uso de esta aplicación reduce la cantidad de información que se transmite y evita la saturación de la red. Además, mitiga el efecto del módulo de visualización sobre el hardware de las estaciones de monitorización donde se encuentre instalado el sistema de vigilancia, por lo que facilitaría su despliegue. El estudio de las principales soluciones a nivel nacional e internacional no permitió encontrar un sistema que produjera estos resultados para Xilema Suria. El sistema fue desarrollado utilizando la metodología RUP (*Rational Unified Process*), el lenguaje de programación C++ y el *framework* de desarrollo Qt. Luego de su implementación, se probó que el software desarrollado cumple con la calidad requerida para ser integrado al sistema de video vigilancia Xilema Suria.

Palabras clave: multiplexor, video, video vigilancia.

Abstract

Xilema Suria product is a video surveillance system supported on IP technology. Its base is formed by simultaneous monitoring stations in real time from different video streams generated by each of the security cameras. The development of an IP Video Multiplexer for Xilema Suria system aims to achieve an application to multiplex video streams transmitted by IP cameras. The use of this application reduces the amount of information transmitted and prevents network congestion. Besides, it mitigates the effect of the display module on hardware monitoring stations where the monitoring system is installed, therefore would facilitate its deployment. The study of both solutions nationally and internationally did not allow to find a system to produce these results for Xilema Suria. The system was developed using the RUP methodology (Rational Unified Process), the programming language C++ and the Qt development framework. After implementation, it was proved that the developed software has the quality required to be integrated to video surveillance system Xilema Suria.

Keywords: multiplexer, video, video surveillance.

¹ **Internet Protocol o Protocolo de Internet (IP):** Es un protocolo de comunicación de datos digitales clasificado funcionalmente en la Capa de Red según el modelo internacional OSI.

Índice General

Índice General

Introducción	1
Capítulo 1: Fundamentación Teórica	5
1.1 Términos asociados al dominio de la investigación.....	5
1.1.1 ¿Qué es un multiplexor de video IP?	6
1.2 Descripción del objeto de estudio	6
1.2.1 Transmisión de flujos de video provenientes de cámaras IP.....	7
1.2.2 Captura de flujos de video provenientes de cámaras IP.	9
1.3 Características del sistema de video vigilancia Xilema Suria	10
1.4 Análisis de soluciones existentes.....	12
1.4.1 Multiplexores de Video IP existentes en Cuba	12
1.4.2 Multiplexores de Video IP existentes en el mundo	14
1.4.3 Conclusiones sobre las Soluciones Existentes	15
1.5 Herramientas y tecnologías para el desarrollo del software	16
1.5.1 Metodologías de Desarrollo	16
1.5.2 Lenguaje de Modelado Unificado (UML).....	17
1.5.3 Herramientas CASE.....	18
1.5.4 Lenguaje de Programación y Biblioteca.....	18
1.5.5 Marco de trabajo.....	20
1.5.6 Entorno de Desarrollo Integrado.....	21
1.5.7 Tecnología para la comunicación ZMQ 4.0.3.....	22
Conclusiones parciales	22
Capítulo 2: Análisis y Diseño del Sistema	23

Índice General

2.1	Modelo de dominio.....	23
2.1.1	<i>Descripción del flujo del diagrama del modelo de dominio.....</i>	23
2.1.2	<i>Diagrama de Modelo de dominio</i>	24
2.1.3	<i>Descripción de las clases</i>	24
2.2	Especificación de los requisitos de software	24
2.2.1	<i>Requisitos Funcionales.....</i>	25
2.2.2	<i>Requisitos No Funcionales</i>	26
2.3	Descripción del sistema.	27
2.3.1	<i>Definición de los actores.....</i>	28
2.3.2	<i>Listado de los Casos de Uso.</i>	28
2.3.3	<i>Diagrama de Casos de Uso del Sistema.....</i>	28
2.3.4	<i>Especificación de los Casos de Uso del Sistema.....</i>	29
2.4	Descripción de la Arquitectura	31
2.4.1	<i>Arquitectura Cliente Servidor</i>	31
2.4.2	<i>Arquitectura en Capas</i>	33
2.5	Patrones de diseño.....	34
2.6	Modelo del diseño.....	36
2.6.1	<i>Diagrama de secuencia del diseño</i>	37
2.6.2	<i>Clase del diseño</i>	37
2.6.3	<i>Diagrama de Clases del Diseño.....</i>	38
	Conclusiones parciales	39
	Capítulo 3: Implementación y Prueba	40
3.1	Modelo de Implementación	40

Índice General

3.1.1	<i>Diagrama de Despliegue</i>	40
3.1.2	<i>Diagrama de componentes de implementación</i>	41
3.2	Pruebas de Software	42
3.2.1	<i>Prueba de Integración</i>	43
3.2.2	<i>Prueba de Sistema</i>	44
3.2.3	<i>Prueba de Aceptación</i>	48
	Conclusiones parciales	49
	Conclusiones Generales	50
	Recomendaciones	51
	Referencias Bibliográficas.....	52
	Bibliografía.....	56
	Anexos.....	57
I.	Entrevista realizada	57
II.	Descripción del Caso de Uso Detener Multiplexor	58
III.	Diagrama de secuencia del Caso de Uso Detener Multiplexor.....	59
IV.	Diagrama de Clases del Diseño.....	59
V.	Acta de Aceptación	60
	Glosario de Términos.....	61

Índice de Figuras

Índice de Figuras

Fig. 1: Modelo de dominio multiplexor de flujos de video provenientes de cámaras IP para el sistema de video vigilancia Xilema Suria.	24
Fig. 2: Diagrama de CU multiplexor de flujos de video provenientes de cámaras IP.	29
Fig. 3: Representación de la Arquitectura Cliente Servidor.	32
Fig. 4: Diagrama de Secuencia CU Transmitir video.	37
Fig. 5: Representación del Diagrama de Clases del Diseño.	38
Fig. 6: Diagrama de Despliegue.	41
Fig. 7: Diagrama de Componentes de Implementación.	42
Fig. 8: Representación de la Integración Incremental Descendente.	43
Fig. 9: Consumo de RAM (MB) en el sistema Xilema Suria, con y sin el uso del multiplexor.	47
Fig. 10: Consumo de CPU (%) en el sistema Xilema Suria, con y sin el uso del multiplexor.	47
Fig. 11: Consumo de Red (Mbps) en el sistema Xilema Suria, con y sin el uso del multiplexor.	48
Fig. 12: Diagrama de Secuencia del CU Detener Multiplexor.	59
Fig. 13: Diagrama de Clases del Diseño.	59
Fig. 14: Acta de Aceptación del Multiplexor de video IP por parte del proyecto de video vigilancia Xilema Suria.	60

Índice de Tablas

Índice de Tablas

Tabla 1: Actores del sistema.....	28
Tabla 2: Descripción del CU Transmitir video.	29
Tabla 3: Resultados de la Prueba de Carga aplicada al multiplexor de video IP.	45
Tabla 4: Resultados de la Prueba de Estrés aplicada al multiplexor de video IP.....	45
Tabla 5: Resultados de las Pruebas de Rendimiento aplicadas al módulo Visor de Xilema Suria.....	46
Tabla 6: Descripción del CU Detener multiplexor.	58

Introducción

Introducción

El término multiplexión tiene su origen en la telegrafía en la década de 1870, pero no es hasta 1910 que George Owen Squier² logra, con el desarrollo del multiplexor telefónico, poner en práctica dicho concepto. Un multiplexor es un dispositivo que recibe tráfico de información de muchas fuentes en líneas de acceso diferentes y las envía a su destino en una única salida (Nassar, 1998).

El amplio desarrollo alcanzado en los últimos años en la informática, las telecomunicaciones y el surgimiento de Internet, ha traído consigo un aumento considerable en la cantidad de información que se transmite a través de las redes de datos. Como consecuencia se genera saturación en las redes, lo que provoca retraso para recibir la información, así como la pérdida de la misma. Tanto tráfico de información también ha provocado inestabilidad para compartir los recursos de la red entre muchos flujos y realizar eficientemente el envío de contenido de uno a muchos clientes (Sun, y otros, 2000).

Como solución a estos problemas se introduce la multiplexión en el campo de la informática, siendo de gran utilidad en áreas como la telefonía, la transmisión digital y el procesamiento de video. Por emplear transmisión de flujos de video en tiempo real, los sistemas de video vigilancia se han beneficiado con el uso de esta técnica, constituyendo un avance importante la creación en 1990 del multiplexor digital que permitió que varias cámaras grabaran al mismo tiempo. El uso de estos dispositivos en un sistema de video vigilancia disminuye el tráfico de datos por la red, además, brinda la posibilidad de unir en un solo flujo de video las transmisiones provenientes de varias cámaras (Bates, y otros, 2007).

Xilema Suria es un sistema de video vigilancia soportado sobre tecnología IP, desarrollado en el Departamento de Componentes del Centro de Geoinformática y Señales Digitales (GEySED) de la Facultad 6 en la Universidad de las Ciencias Informáticas. Su base está formada por estaciones de monitoreo simultáneo y en tiempo real de los diferentes flujos de video generados por cada una de las cámaras de seguridad. Este sistema está formado por siete módulos, siendo los principales el Gestor, con la función de orquestar los procesos que ocurren en el sistema, así como la comunicación entre los diferentes módulos; y el Visor, responsable de visualizar los flujos provenientes de las cámaras IP y servir de interfaz para las funcionalidades que brindan los restantes módulos.

² **George Owen Squier:** científico y militar norteamericano que escribió y editó numerosos libros y artículos sobre el tema de la radio y la electricidad. Su mayor contribución fue la creación del multiplexor telefónico en 1910 que le valió para ser elegido miembro de la Academia Nacional de Ciencias en 1919 (Kennelly, 2010).

Introducción

Actualmente, para visualizar el video proveniente de una cámara IP en el sistema Xilema Suria, el Visor necesita reservar espacio en memoria RAM³ para almacenar las imágenes que componen el flujo emitido por la cámara IP, y aunque el espacio que ocupa la imagen en memoria se libera luego de ser mostrado, pueden llegar a acumularse varias imágenes. Cada una de estas imágenes necesita ser decodificada para su posterior visualización en el Visor, lo cual necesita cálculos matemáticos que son realizados por el CPU⁴. Para visualizar un conjunto de cámaras simultáneamente se utilizan visores independientes en el módulo de monitorización, lo que conlleva una alta utilización de los recursos de hardware. Por ejemplo, si se visualizan a la vez veinte flujos de video se podrían ocupar completamente los recursos de RAM y CPU del nodo de procesamiento.

Además, las cámaras de seguridad pueden encontrarse en redes diferentes y por lo general establecen una conexión *Unicast*⁵ con el sistema, por lo cual se crean múltiples conexiones a cada cámara, siendo un caso frecuente que se formen “cuellos de botella⁶” en las salidas de los dispositivos de interconexión o se sobrepase la cantidad de conexiones que estas admiten. Teniendo en cuenta que cada cámara emite un flujo de video independiente, mientras más cámaras estén visualizándose mayor será el consumo de ancho de banda, llegando a colapsar la red o las capacidades de la estación de monitorización.

La situación descrita anteriormente lleva a plantearse el siguiente **problema a resolver**: ¿Cómo reducir el consumo de recursos de RAM, CPU y red durante la captura y representación de los flujos de video provenientes de cámaras IP en el sistema Xilema Suria? Para realizar un estudio de la problemática y desarrollar la investigación se define como **objeto de estudio**: los procesos de captura y transmisión de flujos de video provenientes de cámaras IP. Para dar solución a la situación problemática descrita y al problema planteado se define como **objetivo general**: desarrollar una aplicación capaz de multiplexar los flujos de video provenientes de cámaras IP, que permita reducir el consumo de recursos de RAM, CPU y red en el sistema Xilema Suria, mediante la captura, multiplexión y transmisión de los videos en tiempo

³ **Random Access Memory (RAM)**: Memoria de Acceso Aleatorio.

⁴ **Central Processing Unit (CPU)**: Unidad Central de Procesos.

⁵ **Unicast**: Conexión donde el cliente establece un enlace con el servidor y recibe o envía una cadena de datos independiente a la comunicación que se realiza con otros clientes.

⁶ **Cuellos de botella (bottleneck)**: Es cuando se realizan muchas solicitudes pero no pueden ser atendidas al mismo tiempo, quedando en una fila de espera hasta llegar un punto que quien está atendiendo las solicitudes no puede atender más ninguna, saturándose y terminando el proceso.

Introducción

real, enmarcándose como **campo de acción**: el proceso de captura y transmisión de video desde cámaras IP dentro del sistema de video vigilancia Xilema Suria.

Para guiar la investigación se definen las siguientes **preguntas científicas**:

- ✓ ¿Cómo reducir el consumo de recursos de RAM, CPU y red en un sistema de video vigilancia?
- ✓ ¿Qué técnicas utilizar para reducir la cantidad de transmisiones que debe procesar un sistema de video vigilancia?
- ✓ ¿Qué tecnologías se deben utilizar para multiplexar los flujos de video emitidos por las cámaras IP?
- ✓ ¿Qué técnicas utilizar para comprobar el uso de recursos en un sistema de video vigilancia?

Con el fin de dar cumplimiento a los objetivos anteriormente planteados, se definieron las siguientes **tareas de investigación**:

1. Establecer los fundamentos teórico-metodológicos para el desarrollo una aplicación capaz de multiplexar los flujos de video provenientes de cámaras IP.
2. Caracterizar los procesos de captura y transmisión de video en lo relativo a la reducción del consumo de recursos dentro de un sistema de video vigilancia.
3. Desarrollar un sistema informático que permita reducir el consumo de recursos al capturar los flujos de video provenientes de cámaras IP en el sistema Xilema Suria.
4. Validar la reducción en el consumo de recursos de RAM, CPU y red durante la captura de los flujos de video provenientes de cámaras IP en el sistema Xilema Suria.

Los **métodos científicos** de investigación que se utilizan con el fin de dar cumplimiento a los objetivos de este trabajo son:

Métodos Teóricos

1. **Analítico-Sintético**: Se emplea para realizar un estudio de la bibliografía referente a los diferentes conceptos asociados al tema, permitiendo la extracción de los elementos más importantes que se relacionan con los procesos de captura y transmisión de flujos de video provenientes de cámaras IP. Se estudiarán los sistemas de video vigilancia y sus características, así como el proceso de transmisión de los flujos de video.

Introducción

- 2. Análisis histórico-lógico:** Este método permite estudiar posibles soluciones que permitan reducir el consumo de recursos en un sistema de video vigilancia; sus antecedentes y su evolución a lo largo de la historia, específicamente se estudiarán las empleadas para capturar y transmitir flujos de video.
- 3. Modelación:** Este método se utiliza fundamentalmente para modelar los artefactos que se generan a lo largo de todo el proceso de desarrollo de software de la herramienta. Por ejemplo, se utiliza en la modelación del diagrama de casos de uso, el diagrama de clases y el diagrama de despliegue del multiplexor de video IP.

Métodos Empíricos

- 1. Entrevista:** Se utiliza con el objetivo de entrevistar a los miembros del proyecto Xilema Suria para definir las principales funcionalidades que debe cubrir la aplicación a implementar. En el encuentro con los clientes se determina la importancia del desarrollo de la aplicación, así como los beneficios que aportaría al proyecto.

El presente trabajo está distribuido en los siguientes capítulos:

Capítulo 1: Fundamentación teórica. En este capítulo se fundamentan términos de importancia para la investigación. Se realiza un estudio del arte sobre sistemas que existen en Cuba y el mundo con características similares a las requeridas para la investigación. Además se analiza el objeto de estudio y se seleccionan las tecnologías, los lenguajes de programación, las herramientas y la metodología para el desarrollo de la solución.

Capítulo 2: Análisis y diseño del sistema. En este capítulo se describe el modelo del dominio del sistema que se va a desarrollar. También se identifican los requisitos funcionales y no funcionales. Se definen los casos de usos y actores del sistema. Se especifica la arquitectura base de la aplicación, mostrándose los artefactos generados en esta fase así como los patrones de diseño utilizados.

Capítulo 3: Implementación y prueba. En este capítulo se aborda el desarrollo de la solución, mostrándose los artefactos generados por la metodología utilizada. Se muestran el diagrama de despliegue y el de componentes de implementación. También se realizan las descripciones de las pruebas y se exponen los resultados de las mismas.

Capítulo 1: Fundamentación Teórica

Capítulo 1: Fundamentación Teórica

Introducción

Este capítulo tiene como objetivo realizar una introducción al tema de la investigación, haciendo referencia a conceptos asociados al dominio del problema a resolver. Se realiza un estudio acerca de los multiplexores de video IP existentes en Cuba y el mundo. Además se mencionan las principales herramientas y tecnologías que serán utilizadas en el desarrollo de la aplicación.

1.1 Términos asociados al dominio de la investigación

Video vigilancia

Los sistemas de video vigilancia como Xilema Suria interactúan con un conjunto de cámaras. Estas se encuentran interconectadas mediante cableado o inalámbricamente a una estación central o servidor para la administración del sistema y el almacenamiento de datos. El proceso de vigilancia consiste en monitorizar una secuencia de video emitido por una cámara analógica o una cámara IP (Kruegle, 2011).

Cámara IP

El uso de cámaras IP para video vigilancia se conoce como video vigilancia IP, siendo una efectiva solución de seguridad que ofrece monitorización y control avanzado en sistemas de seguridad. Son dispositivos autónomos que cuentan con un servidor de video incorporado que les permite transmitir las imágenes capturadas a través de redes IP. Las cámaras IP permiten al usuario tener la cámara en una localización y ver el video digital, en tiempo real, desde otro lugar (Valeriano, 2010). La presente investigación centra su enfoque en este tipo de cámaras debido a que Xilema Suria es un sistema de video vigilancia que opera sobre redes digitales, por lo tanto, los flujos de video digital procesados en el sistema provienen de cámaras IP.

Video digital

El video digital consiste en mostrar una sucesión de imágenes digitales. Estas imágenes se muestran a una frecuencia determinada, lo que hace posible saber el número de bytes mostrados(o transmitidos) por unidad de tiempo. Dichos videos pueden ser tomados utilizando dispositivos como las cámaras IP, o pueden ser creados directamente en una computadora mediante el uso de programas especializados. En un sistema de video vigilancia IP, representan la señal de video emitida por las cámaras y es el resultado

Capítulo 1: Fundamentación Teórica

final que es mostrado en las estaciones de monitorización, grabado para su posterior análisis o procesado en busca de información de interés para el usuario (Montagu, 2010).

Multiplexión

Como se plantea anteriormente, procesar cada flujo de video entrante en el sistema Xilema Suria conlleva un alto consumo de recursos. Atendiendo a esto, la presente investigación se centra en hacer llegar a las estaciones de monitorización el video proveniente de múltiples cámaras IP en un mismo flujo. Para lograr la transmisión de varias señales independientes por un mismo canal se utiliza la multiplexión, la cual aumenta la eficiencia en las redes de comunicación (Forouzan, 2008), por lo que se arribó a la conclusión de que el desarrollo de un multiplexor de video IP solucionaría los problemas existentes en el sistema Xilema Suria, en lo referente a la captura y visualización simultánea de un gran número de cámaras de seguridad.

1.1.1 ¿Qué es un multiplexor de video IP?

Teniendo en cuenta las definiciones anteriores, se plantea que un multiplexor de video IP es una aplicación informática que construye un nuevo flujo de video a partir de otros provenientes de cámaras IP. El flujo de video resultante muestra en una posición predefinida los flujos de video de las cámaras IP solicitadas, permitiendo integrar varios videos en uno solo. Realizar este proceso conlleva una disminución en el consumo de red al transmitir sobre una red LAN⁷.

El funcionamiento de un multiplexor de video IP se divide en las siguientes etapas:

1. Captura de los flujos de video provenientes de las cámaras IP solicitadas.
2. Creación de un nuevo video conformado por los flujos de video obtenidos, ubicados uniformemente en una misma rejilla.
3. Transmisión del video resultante hacia las estaciones de monitorización.

1.2 Descripción del objeto de estudio

Para la realización de esta investigación es necesario profundizar en los procesos de captura y transmisión de flujos de video provenientes de cámaras IP. Su estudio permite una mejor comprensión de la función que realizan dentro de un sistema de video vigilancia.

⁷ **Local Area Network (LAN):** Red de Área Local.

Capítulo 1: Fundamentación Teórica

1.2.1 Transmisión de flujos de video provenientes de cámaras IP

El acceso de los usuarios a Internet y la posibilidad de adquirir hardware cada vez más potente, son factores que impulsan el proceso de difusión de contenidos multimedia. Con el surgimiento y evolución de la red de redes, la transmisión de materiales audiovisuales en tiempo real y bajo demanda ha marcado un hito en la historia de los medios de difusión. Resulta común en la actualidad que los usuarios disfruten de la posibilidad de visualizar y escuchar archivos de audio y video desde cualquier parte del mundo con solo contar con un dispositivo con acceso a Internet.

La tecnología *streaming* permite la comunicación entre diferentes usuarios, la interactividad y el procesamiento de la información de forma rápida y sin interrupciones. Esta tecnología de transferencia de datos posibilita la transmisión de videos y mensajes audiovisuales en la red sin que se cargue completamente la información. De esta manera es posible ver el contenido antes de terminar la transmisión. Esto ofrece la posibilidad de avanzar hasta el punto que se desee del mismo, ajustándose a las características actuales, como la velocidad de conexión o el tráfico en la red (Bernal, 2010).

La filosofía del *streaming* consiste en dividir un archivo grande en pequeños paquetes que se pueden enviar de forma continua, para que el destinatario comience a reproducir el archivo cuando su *buffer* de entrada esté lleno. Los paquetes son almacenados solamente el tiempo necesario para reproducirlos, por lo que estos no quedan almacenados en el disco duro (Novoa, 2007).

Tipos de *streaming*

El proceso de *streaming* se puede dividir en dos categorías en función de cómo se obtiene la información a difundir: *streaming* en directo o en vivo y *streaming* bajo demanda.

- **El *streaming* en directo o en vivo** es aquel que transmite eventos que están sucediendo justo en el momento de la difusión. Por ejemplo, la transmisión de conciertos o clases, son eventos que típicamente se difunden usando este tipo de *streaming*. La transmisión de radio y televisión por Internet también tienen estas características, aunque en ocasiones, la información que se difunde no parte de un evento en directo (Novoa, 2007). Las cámaras IP envían sus flujos de video a través del *streaming* en vivo. En la transmisión de los flujos de video desde la aplicación hacia las estaciones de monitorización se utilizará esta categoría.
- **El *streaming* bajo demanda**, permite obtener acceso a los contenidos audiovisuales de forma personalizada, ofreciendo la posibilidad de solicitar y visualizar un material concreto en el momento

Capítulo 1: Fundamentación Teórica

exacto que el telespectador lo desee, por lo que no existe la posibilidad de observar dicho material en tiempo real. Las medias a transmitir deben haber sido grabadas con anterioridad, para comenzar su reproducción cuando el cliente lo solicite (Torres, 2009).

Protocolos empleados para emitir *streaming*

Un protocolo es un conjunto de normas o estándares que especifican el método para enviar y recibir datos entre varios ordenadores, controlando o permitiendo la conexión, comunicación y transferencia de la información entre ellos (Santos, 2007). Dentro de la video vigilancia, los principales protocolos utilizados en la transmisión de los flujos de video provenientes de las cámaras IP son:

Protocolo RTSP

El protocolo *Real Time Streaming Protocol* (RTSP) establece y controla uno o varios flujos sincronizados de datos, ya sean de audio o de video. Este protocolo de flujo de datos en tiempo real actúa como un mando a través de la red para servidores multimedia y se sitúa en el nivel de aplicación dentro del modelo OSI⁸. Al ser un protocolo no orientado a conexión, en el transcurso de una sesión RTSP un cliente puede abrir y cerrar varias conexiones a nivel de transporte, sin que dicha sesión se vea afectada. RTSP es únicamente un protocolo de control, por lo que el envío de datos de *streaming* se deberá realizar mediante algún otro protocolo destinado a tal efecto (Novoa, 2007). Este protocolo es empleado por las cámaras IP soportadas por el sistema Xilema Suria. El multiplexor de video IP transmite el flujo de video multiplexado hacia las estaciones de monitorización a través de este protocolo ya que el módulo Visor de Xilema Suria recibe flujos de video RTSP.

Protocolo HTTP

Hypertext Transport Protocol (HTTP) es el protocolo usado en cada transacción de la web. Mediante él, se envían las peticiones de acceso a una página, la respuesta con el contenido y también información en ambos sentidos. Específicamente en la transmisión de video, este protocolo es usado para que un servidor web pueda enviar los datos a un medio visualizador, permitiéndole al cliente la descarga automática (Santos, 2007). Algunos modelos de cámaras IP incorporan este protocolo para transmitir los flujos de video capturados.

⁸ **Open System Interconnection (OSI)**: modelo de interconexión de sistemas abiertos (ISO/IEC 7498-1).

Capítulo 1: Fundamentación Teórica

Protocolo UDP

El protocolo UDP (*User Datagram Protocol*) es un protocolo que provee las funciones básicas de la capa de transporte. Tiene una sobrecarga mucho menor que el protocolo TCP, ya que no está orientado a la conexión y no proporciona mecanismos sofisticados de retransmisión y flujo de control. Dicho protocolo proporciona una comunicación sencilla entre las aplicaciones de dos ordenadores. Este protocolo de transporte suele ser usado en aplicaciones de *streaming* (video o audio), ya que en estas es más importante la recepción rápida de los datos que la verificación de los mismos (Novoa, 2007). Este protocolo será utilizado en la transmisión del flujo de video multiplexado hacia las estaciones de monitorización. La prioridad del sistema es que el video sea entregado con el menor retardo posible, independientemente de que pueda existir pérdida de algunos paquetes.

1.2.2 Captura de flujos de video provenientes de cámaras IP.

La captura de un flujo de video es el proceso mediante el cual se obtiene el video proveniente de un dispositivo que se encuentre transmitiendo, ya sea una transmisión en tiempo real o bajo demanda (Schulzrinne, y otros, 1998).

Captura de un flujo de video RTSP

El proceso de obtención de un flujo de video vía RTSP se basa en peticiones cliente-servidor. A continuación se describe como se realiza este proceso (Schulzrinne, y otros, 1998):

1. El cliente accede a la URL⁹ RTSP utilizando el nombre del servidor, el puerto y el nombre de acceso al flujo de video específico que desea capturar. También la URL RTSP puede contener, en caso de ser necesario, usuario y contraseña para acceder al flujo.
2. El cliente realiza una conexión con el servidor, utilizando el protocolo TCP.
3. Cuando la conexión está establecida correctamente, el cliente envía al servidor una petición OPTIONS con datos necesarios para obtener el flujo.
4. EL servidor devuelve información que incluye la fecha, el número de sesión, el nombre del servidor y los métodos soportados.

⁹ **Uniform Resource Locator (URL):** Localizador de Recursos Uniforme. Es un identificador formado por una secuencia de caracteres, de acuerdo a un formato y estándar, que designa recursos en una red (Masinter, y otros, 2010).

Capítulo 1: Fundamentación Teórica

5. El cliente envía una petición DESCRIBE para obtener una descripción del flujo. El servidor responde con todos los valores necesarios para la inicialización.
6. El cliente envía una petición SETUP para el flujo de datos que se quiere reproducir. El SETUP especifica los protocolos aceptados para el transporte de los datos del video.
7. El cliente inicializa los programas requeridos para reproducir el flujo o procesarlo.
8. El cliente envía una petición PLAY para que el servidor comience a transmitir el video.
9. Cuando el cliente ha terminado con el flujo de video, envía un SET_PARAMETER que contiene las estadísticas de la sesión, seguido de una petición TEARDOWN para dar por terminada la conexión con el servidor.

Captura de un flujo de video HTTP

Para capturar un flujo de video vía HTTP:

1. El cliente accede al flujo de video utilizando una dirección URL que contiene el identificador del servidor así como el puerto por el que se está transmitiendo. En caso de necesitar autenticación, la URL también contendrá el usuario y contraseña para establecer la conexión.
2. El cliente realiza una conexión con el servidor, utilizando el protocolo TCP.
3. El cliente obtiene un archivo con varias etiquetas, en este caso debe verificar que el archivo comienza con la etiqueta EXTM3U y que la etiqueta EXT-X-VERSION especifica una versión soportada. Si no soporta la versión especificada entonces no se puede reproducir o procesar el flujo de video.
4. Si el cliente soporta la versión especificada por el servidor, entonces se analiza la etiqueta EXTINF que especifica el tiempo en milisegundos para acceder periódicamente al flujo de video.
5. El cliente inicializa los programas requeridos para reproducir el flujo de video o procesarlo.
6. El cliente obtiene el flujo de video periódicamente desde el servidor.
7. Si el cliente no hace peticiones en un tiempo determinado al servidor, este termina la conexión.

1.3 Características del sistema de video vigilancia Xilema Suria

El producto Xilema Suria se concibe sobre la base del desarrollo de un sistema de video vigilancia soportado sobre tecnología IP. Su base está formada por estaciones de monitoreo simultáneo y en tiempo

Capítulo 1: Fundamentación Teórica

real de los diferentes flujos de video generados por cada una de las cámaras de seguridad. El objetivo que persigue el sistema es proveer una herramienta para video vigilancia adaptable a varios entornos y plataformas. Este software tiene la capacidad de visualizar, grabar, recuperar y analizar los flujos de video generados por las cámaras de seguridad, requiriéndose menos esfuerzo para establecer la protección de los lugares donde sea desplegado el sistema. El sistema se estructura básicamente en siete módulos:

Gestor: Es el módulo fundamental de la aplicación, y todos los demás son agentes que se encargan de realizar tareas específicas. Todo el tráfico de información pasa por el Gestor facilitando el control y supervisión, permitiendo además la flexibilidad del sistema. Es el único que interactúa directamente con la Base de datos.

Visor: Es el módulo con el que interactúan los usuarios. Permite visualizar los flujos de video capturados por las cámaras. Pueden existir varias instancias de este módulo ejecutándose en los dominios físicos del sistema o ninguna. Tiene la capacidad de reflejar todo el aspecto organizativo con que se manejan las cámaras internamente, además de poder visualizarlas de manera independiente o colectiva (a manera de vistas). También permite manipularlas en la medida de las capacidades de cada una.

Algunas funcionalidades que ofrece:

- ✓ Visualizar las cámaras insertadas en el sistema organizadas según su disposición física.
- ✓ Gestionar configuración del Visor.
- ✓ Gestionar modos de trabajo.
- ✓ Realizar peticiones.
- ✓ Visualizar videos obtenidos desde las cámaras y una vez visualizado un video podrá:
 1. Refrescar la visualización del video mostrado.
 2. Adicionar visualizadores.
 3. Ver en modo “Pantalla completa” el video visualizado.
 4. Ver en modo “Visualizador Independiente” el video.
 5. Ajustar visualizador.
 6. Mostrar información de la cámara visualizada.

Capítulo 1: Fundamentación Teórica

Administración: Es el módulo encargado de gestionar los usuarios, roles y permisos en el sistema, así como la planificación de tareas de grabación y análisis.

Servidor de Análisis (*Analytic*): Es el módulo encargado de realizar el procesamiento a los flujos de video que le sean asignados ya sea por planificación o bajo demanda.

Servidor de Grabación: Es el módulo encargado de almacenar los flujos de video obtenidos de las cámaras. La grabación puede comenzar por petición de un cliente determinado o por configuración, que puede ser por horarios determinados o por la ocurrencia de algún evento.

Autonomía: Es el módulo encargado de controlar la capacidad en disco en el almacén de grabaciones. Es capaz de tomar decisiones de eliminar, mover, o copiar archivos, según las reglas definidas por el usuario.

Recuperador: Es el módulo encargado de recuperar los videos almacenados en los servidores. Permite hacer búsquedas en estos, servir los resultados al usuario y darle la posibilidad de la reproducción de los mismos a diferentes niveles de velocidad.

1.4 Análisis de soluciones existentes

Aunque son usados en las telecomunicaciones de forma general, los multiplexores de video son muy utilizados en los sistemas de video vigilancia. Esto se debe a que satisfacen necesidades como la reducción del consumo de red (el cual se ve incrementado debido a la transmisión de múltiples flujos de video al mismo tiempo) y el volumen de procesamiento en el hardware de las estaciones de monitorización.

1.4.1 Multiplexores de Video IP existentes en Cuba

En Cuba, la Empresa de Tecnología e Información para la Defensa (XETID) se encuentra desarrollando un servidor de *streaming*, el cual funciona como mediador entre el Visor web y las cámaras, evitando así alcanzar el máximo número de conexiones permitidas por las cámaras IP. Por ejemplo, si se desea visualizar alguna de las cámaras de vigilancia instaladas, se le hace la petición al servidor de *streaming* y este se encarga de conectarse a la cámara y emitir el flujo de video obtenido. Si múltiples usuarios necesitan acceder a la misma cámara, el servidor mantiene una sola conexión con esta y atiende todas las peticiones realizadas. Esta característica es necesaria en la presente investigación y el estudio de esta

Capítulo 1: Fundamentación Teórica

solución permitió comprender como se realiza este proceso. El servidor de *streaming* de la empresa XETID no constituye una solución viable para Xilema Suria debido a que no es capaz de multiplexar video.

La empresa Desarrollo de Aplicaciones, Tecnologías y Sistemas (DATYS), dedicada al desarrollo de aplicaciones informáticas, cuenta entre sus productos con Xyma Safe Vision, software de video vigilancia profesional basado en tecnología IP que tiene como objetivo el monitoreo en tiempo real y la vigilancia de instalaciones y exteriores. Este software brinda facilidades para la gestión del ancho de banda a través de la personalización de los parámetros del video que se transmite (formato, tamaño, cantidad de cuadros por segundo, tipo de compresión) para cada canal (DATYS, 2011). Otra característica de este sistema es el procesamiento distribuido, o sea, los dispositivos de video IP llevan a cabo una parte importante del procesamiento de imagen, aliviando el tráfico de la red y evitando el procesamiento excesivo en las estaciones de trabajo. En el estudio de esta solución no fue identificada ninguna funcionalidad que pueda servir de base en el desarrollo de la presente investigación. El software Xyma Safe Vision no cuenta con un sistema que se encargue de multiplexar los flujos de video provenientes de las cámaras IP y por tanto no representa una solución viable para Xilema Suria.

El centro de desarrollo GEySED de la Facultad 6 en la Universidad de las Ciencias Informáticas, cuenta con un proyecto encargado de automatizar los procesos de planificación y transmisión de canales virtuales que se ejecutan en una entidad dedicada a la gestión y transmisión de contenidos audiovisuales (STCV). Con el objetivo de responder correctamente a la alta demanda concurrente de reproducción de medias, en el producto STCV se desarrolló un **Subsistema distribuido de transmisión de contenido audiovisual** basado en los principales protocolos y formatos de video para realizar *streaming*. Este sistema permite la transmisión estable de uno o varios canales ante múltiples solicitudes de materiales audiovisuales, evitando inestabilidad en el flujo de datos, así como la mala visualización y audición de las medias transmitidas (Cabrera, y otros, 2013). Este sistema no representa una solución viable ya que no multiplexa video, pero su estudio consolidó el empleo de la biblioteca GStreamer para la edición y transmisión de los flujos de video. Además, determinó el uso del protocolo UDP en la transmisión hacia las estaciones de monitorización del video multiplexado y aportó información sobre el proceso de transmisión de flujos de video (tipos de *streaming*, protocolos empleados, aplicaciones del *streaming*).

A pesar de la creciente evolución del desarrollo de software que experimenta el país, actualmente no existe un uso generalizado de los sistemas de video vigilancia ni de sistemas que multiplexen video. Algunas empresas como XETID y DATYS, ya han incursionado en el tema, pero sus soluciones no

Capítulo 1: Fundamentación Teórica

resultaron factibles para la investigación; aunque sí ofrecieron nuevos conocimientos sobre los procesos de captura y transmisión de los flujos de video provenientes de las cámaras IP, reafirmando el uso de la biblioteca GStreamer para la transmisión y procesamiento de los mismos.

1.4.2 Multiplexores de Video IP existentes en el mundo

A nivel internacional han sido desarrollados diferentes multiplexores, que pueden ser dispositivos de hardware o software. Un dispositivo multiplexor de video permite que las señales de video de múltiples cámaras de seguridad, se puedan combinar y mostrar en un monitor. Esto permite que el video de todas las cámaras se pueda grabar en un dispositivo de grabación que tiene una sola entrada de video, como un reproductor de video vigilancia. Por ejemplo, si se tienen cuatro cámaras de seguridad, cada una de las cuatro señales de video de cada cámara es sincronizada por el multiplexor y grabada en la misma cinta de video (Bond, y otros, 2009). Estos dispositivos son muy útiles en sistemas de video vigilancia ya que permiten visualizar múltiples cámaras simultáneamente. Algunos de estos dispositivos son:

Basscam™ QCS4XL

Visualiza simultáneamente imágenes de 4, 8 o 16 cámaras en un monitor de video. Dicho sistema permite la visualización de múltiples videos escalados utilizando para su creación un único decodificador de video. Varios flujos de datos de video digital se pueden visualizar a tamaño completo, o escalados en una misma pantalla. Para seleccionar uno o más de los videos para su visualización, se utiliza un mando a distancia que permite ver los videos en tiempo real a tamaño completo o con un patrón definido (Bass Electronics Incorporation, 2006). Este dispositivo no representa una solución viable ya que las cámaras se conectan directamente al multiplexor, por lo que no son configurables por el Visor del sistema de video vigilancia Xilema Suria. Su precio en el mercado internacional es de \$ 199.99.

Basscam™ QMS4XL

Multiplexor que permite la selección de la señal de video de cada uno de las múltiples cámaras de televisión conectadas. Simultáneamente muestra 4 flujos de video de diferentes cámaras en cuadrantes diferentes de un monitor. Dicha selección se puede hacer de acuerdo con la importancia o prioridad de la zona en la que se encuentra la cámara, o debido a un cambio de evento en una de ellas. También se puede establecer un tiempo para cada cámara, que permite al sistema mostrar la imagen correspondiente por el tiempo establecido y cambiar, al expirar este, a la próxima señal. Además, se puede reproducir o demultiplexar la señal de video grabada, debido a que el sistema también tiene un medio para insertar en

Capítulo 1: Fundamentación Teórica

dicha señal de video multiplexado señales de control, tales como información de eventos e información del tiempo. Este dispositivo no es viable para el sistema Xilema Suria ya que funciona sobre redes analógicas. Su precio oscila alrededor de los \$ 229.99 (Bass Electronics Incorporation, 2006).

VM-16RT de CCTV Camera Pros

Este sistema está formado por un multiplexor-codificador de video para recibir selectivamente múltiples señales de video, transferir y simultáneamente visualizar las mismas, con un patrón predefinido conformado por varias secciones (una para cada señal), en una pantalla. Además, incluye un decodificador-convertidor que recibe, decodifica y convierte una señal codificada de video multiplexado para la visualización selectiva y simultánea de las múltiples imágenes en un dispositivo. La señal de video multiplexado incluye también datos de control de visualización que representan la posición, el tamaño y la prioridad relativa de cada una de las imágenes de video dentro del patrón de visualización seleccionado. Las cámaras pueden ser visualizadas en diferentes modos tales como: una sola cámara, dos, cuatro u ocho cámaras simultáneamente (Bond, y otros, 2010). Sin embargo, este dispositivo no cuenta con una interfaz de comunicación accesible desde el módulo Visor de Xilema Suria. Además, necesita de la interacción física del usuario para iniciar y detener el servicio de multiplexión.

Cisco IPTV Head-end

Dispositivos desarrollados por Cisco que permiten la adquisición y conversión de video de una amplia gama de fuentes, incluidas cámaras IP y analógicas. Además, cuentan con procesamiento avanzado de señales de video, lo cual incluye la característica de multiplexar video, brindando al usuario la posibilidad de unir varios flujos de video en uno, posibilitando visualizar varias cámaras en una sola pantalla. En este sistema, el usuario define la disposición en que serán mostradas las cámaras, además incluye escalado de imágenes, filtrados de datos, hacer *zoom* y es capaz de mostrar videos en alta definición (Cisco Systems Incorporation, 2010). Su precio se encuentra entre 200 y 300 dólares. No es viable su uso ya que se requeriría la compra de un gran número de estos dispositivos para realizar un correcto despliegue del sistema de video vigilancia Xilema Suria.

1.4.3 Conclusiones sobre las Soluciones Existentes

Luego de una investigación a nivel nacional sobre los multiplexores de video, se puede concluir que a pesar de existir empresas dedicadas al desarrollo de sistemas de video vigilancia, no se encontró una aplicación que multiplexe los flujos de video provenientes de las cámaras IP. A nivel internacional se

Capítulo 1: Fundamentación Teórica

identificaron multiplexores como **Basscam™ QMS4XL**, el cual funciona sobre redes analógicas y no digitales, por lo que no es idóneo para el sistema Xilema Suria. Por otra parte, el análisis de los dispositivos **Cisco IPTV Head-end**, permitió identificar funcionalidades propias del multiplexor de video IP, como la unión de varios flujos de video en uno y la disposición en que se muestran las cámaras. Además, el uso de dichos dispositivos implicaría un costo adicional al desplegar el sistema, ya que tienen un precio elevado en el mercado.

Por todo lo anteriormente planteado, se concluye que las soluciones analizadas no satisfacen las necesidades de Xilema Suria. Por tanto, se requiere crear un software que permita la multiplexión de los flujos de video provenientes de las cámaras IP conectadas al sistema de video vigilancia Xilema Suria.

1.5 Herramientas y tecnologías para el desarrollo del software

Teniendo en cuenta que la aplicación es de interés exclusivo para el proyecto de video vigilancia Xilema Suria, se empleará para su desarrollo las herramientas y las tecnologías que se definen en este proyecto, con el objetivo de lograr una fácil integración con sus módulos.

1.5.1 Metodologías de Desarrollo

Las metodologías de desarrollo de software son el marco de trabajo que colecciona un conjunto de pasos y procedimientos que se deben seguir para organizar, controlar y planear el proceso de desarrollo de un software (Pressman, 2010). Hoy en día existen varias metodologías que brindan soluciones a los miembros de un proyecto. Su uso posee una enorme importancia a la hora de gestionar de forma eficiente un proyecto.

Proceso Unificado de Desarrollo de Software (RUP)

El Proceso Unificado de Desarrollo (RUP) constituye junto al Lenguaje Unificado de Modelado (UML) una de las metodologías más utilizadas para el análisis, implementación y documentación de sistemas orientados a objetos. Es un proceso que puede especializarse para una gran variedad de sistemas de software, en diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. Su objetivo es asegurar la producción de software de alta

Capítulo 1: Fundamentación Teórica

calidad, que satisfaga los requerimientos de los usuarios finales respetando el cronograma y el presupuesto (Jacobson, y otros, 2000).

Principales características (Jacobson, y otros, 2000):

- ✓ **Proceso Dirigido por los Casos de Uso:** Se refiere a la utilización de los Casos de Uso para el desenvolvimiento y desarrollo de las disciplinas con los artefactos, roles y actividades necesarias. Un Caso de Uso es la secuencia de pasos que conlleva la realización e implementación de un requisito planteado por el cliente.
- ✓ **Proceso Iterativo e Incremental:** Es el modelo utilizado por RUP para el desarrollo de un proyecto de software. Este modelo plantea la implementación del proyecto a realizar en iteraciones, con lo cual se pueden definir objetivos por cumplir en cada iteración y así poder ir completando todo el proyecto iteración por iteración. Esto permite tener pequeños avances del proyecto que son entregables al cliente, el cual puede probar mientras se está desarrollando otra iteración del producto.
- ✓ **Centrado en la arquitectura:** RUP presta especial atención al establecimiento temprano de una buena arquitectura, con el objetivo de evitar un fuerte impacto ante cambios posteriores durante la construcción y mantenimiento. La arquitectura muestra la visión común del sistema completo en la que tanto el equipo de proyecto como los usuarios deben estar de acuerdo.

1.5.2 Lenguaje de Modelado Unificado (UML).

EL Lenguaje de Modelado Unificado (UML) es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de base de datos y componentes de software reutilizables (Jacobson, y otros, 2005). El uso de UML en esta investigación permite aprovechar las siguientes ventajas:

- ✓ Permite especificar, visualizar y documentar los artefactos que son generados a lo largo del ciclo de desarrollo de la aplicación, fundamentalmente en las fases de análisis, diseño e implementación.
- ✓ Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software, lo cual posibilita que se pueda usar como lenguaje de modelado en la metodología RUP.

Capítulo 1: Fundamentación Teórica

1.5.3 Herramientas CASE¹⁰

Las herramientas CASE son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante los pasos del ciclo de vida de desarrollo de un software. Las herramientas CASE también permiten a los analistas tener más tiempo para el análisis y diseño, así como disminuir el tiempo para codificar y probar (Case, 2001).

Visual Paradigm 8.0

Con el fin de automatizar los aspectos claves del proceso de desarrollo del multiplexor de video IP para el sistema Xilema Suria, se hace necesaria la utilización de las herramientas CASE. Para el desarrollo de esta investigación se utilizará la herramienta Visual Paradigm 8.0. Algunas de sus características son (Case, 2001):

- ✓ Disponibilidad en múltiples plataformas, lo cual facilita que la realización de diagramas de modelado no dependan del sistema operativo que se tenga instalado en el equipo de trabajo.
- ✓ Uso de un lenguaje estándar que es común a todo el equipo de proyecto, facilitando la comunicación, permitiendo la realización de diagramas y modelos durante el proceso de desarrollo.
- ✓ Licencia: gratuita y comercial, fácil de instalar y actualizar.

1.5.4 Lenguaje de Programación y Biblioteca

Un lenguaje de programación actúa como un traductor entre el usuario y el equipo. En lugar de aprender el lenguaje nativo del equipo (conocido como lenguaje máquina o de bajo nivel), se puede utilizar un lenguaje de programación de alto nivel para dar instrucciones al equipo de un modo que sea más fácil de aprender y entender. Pueden usarse para crear programas que controlen el comportamiento de una máquina o para expresar algoritmos con precisión. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones (Microsoft Corporation, 2011).

C++

C++ es una versión ampliada del lenguaje C, incluyendo además algunas mejoras considerables como el soporte de la programación orientada a objetos. Es un lenguaje potente que mantiene las ventajas de C

¹⁰ **Computer Aided Software Engineering (CASE):** Ingeniería de Software Asistida por Computación.

Capítulo 1: Fundamentación Teórica

en cuanto a riqueza de operadores, expresiones, flexibilidad y eficiencia. El código es portable y puede ejecutarse en cualquier máquina y bajo cualquier sistema operativo. El estándar de C++ utilizado es el **ISO/IEC 14882:1998**. La elección de este lenguaje se debe a las siguientes características (C++, 2014):

- ✓ **Chequeo de Tipos:** Lenguaje de tipificación fuerte, pone restricciones sobre cómo diferentes tipos de variables se pueden convertir entre sí. Tipificación manifiesto, los tipos de variables se definen explícitamente. Soporta la revisión del tipo estático y dinámico. C++ permite conversiones de tipos para ser revisados, ya sea en tiempo de compilación o en tiempo de ejecución.
- ✓ **Programación Orientada a Objetos:** Permite expresar programas en los términos de "objetos", que están destinados a modelar objetos en el mundo real. Tal paradigma permite que el código sea reutilizado de manera notable y sea más fácil de entender además de incrementar la productividad y calidad del software.
- ✓ **Eficiencia:** Es uno de los lenguajes más rápidos en cuanto a tiempo de ejecución ya que permite la separación de un programa en módulos que admiten compilación independiente, ventaja que puede ser aprovechada para el procesamiento de los flujos de video. Además, incluye un uso extensivo de apuntadores para la memoria, arreglos, estructuras y funciones, al mismo tiempo que es capaz de manejar actividades de bajo nivel y generar programas eficientes.
- ✓ **Herramientas:** Existe una gran cantidad de compiladores, depuradores y bibliotecas que proporcionan un conjunto extenso de capacidades de entrada/salida.

Biblioteca *GStreamer* 1.0

Las bibliotecas son conjuntos de subprogramas que contienen datos y códigos que brindan servicios a otras aplicaciones y ayudan a los programadores en el desarrollo de software. Estas no necesitan ser modificadas y el código que contienen se añade al programa principal cuando se genera.

GStreamer es una plataforma multimedia basado en tuberías (*pipelines*), escrito en el lenguaje de programación C, con el sistema de tipos basado en GObject. El diseño en *pipelines* sirve como base para crear aplicaciones multimedia como editores de vídeo, emisoras de radiodifusión y reproductores

Capítulo 1: Fundamentación Teórica

multimedia. *GStreamer* es multiplataforma y está patentada bajo la licencia GPL¹¹ (Taymans, y otros, 2012).

Características de *GStreamer*:

- ✓ Incluye una API¹² para aplicaciones multimedia.
- ✓ Permite mezclar diferentes tipos de medios como pueden ser audio, video, imágenes y subtítulos.
- ✓ Posibilita aplicar diferentes tipos de filtros, tanto de audio como de video.
- ✓ Facilita la transmisión y recepción de datos en tiempo real.
- ✓ Posibilita la reproducción de archivos multimedia.

Utilizando *GStreamer* se logra garantizar funcionalidades básicas y útiles del sistema, como hacer composiciones de video utilizando varias fuentes. Provee un mecanismo por el cual se puede especificar la posición en la que se muestra cada fuente.

1.5.5 Marco de trabajo

Un marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que permite resolver nuevos problemas. En el desarrollo de software constituye una estructura conceptual y tecnológica de soporte definida. Normalmente, cuenta con artefactos o módulos de software concretos, que se emplean fundamentalmente para desarrollar código fuente (Quesada, 2008).

Qt 5.3

Marco de trabajo multiplataforma de código abierto que se utiliza generalmente para desarrollar aplicaciones de software haciendo uso de una interfaz gráfica GUI¹³. Cuenta con soporte para la programación concurrente y la comunicación con bases de datos. Qt utiliza C++ como lenguaje de programación nativo, haciendo un uso extensivo de un generador de código llamado MOC¹⁴, junto con

¹¹ **General Public License (GPL)**: Licencia que otorga a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

¹² **Application Programming Interface (API)**: Interfaz de Programación de Aplicaciones.

¹³ **Graphical User Interface (GUI)**: Interfaz Gráfica de Usuario.

¹⁴ **Meta Object Compiler (MOC)**: Compilador de Meta Objetos.

Capítulo 1: Fundamentación Teórica

varias macros para enriquecer el lenguaje (Qt, 2014). Este marco de trabajo fue seleccionado debido a que cumple con las políticas definidas para el proyecto Xilema Suria en el documento de arquitectura, haciendo uso de las ventajas que ofrecen las tecnologías libres.

1.5.6 Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado (por sus siglas en inglés, IDE) es un programa que está compuesto por un conjunto de herramientas para un programador. Facilita un marco de trabajo amigable para una gran cantidad de lenguajes de programación tales como C++, Java, C#, logrando utilizarse en el mismo uno o varios lenguajes de programación (Fritzler, 2011).

Qt Creator 3.0

En esta investigación se utilizará el IDE *Qt Creator 3.0* para el desarrollo del multiplexor de video IP. Esta elección se debe a que proporciona amplios beneficios para los desarrolladores debido a que existe abundante documentación sobre cómo trabajar en su entorno, ayudando a los nuevos usuarios de Qt a aprender y comenzar a desarrollar rápidamente. Además, era necesaria la utilización de un IDE que aprovechara toda la potencialidad del lenguaje C++ y que ofreciera características tales como (Qt Project, 2013):

- ✓ Las bibliotecas Qt: clases escritas en C++ que facilitan el desarrollo.
 - **QtDesigner**: para diseñar formularios visualmente.
 - **QtAssistant**: brinda acceso rápido a la documentación.
 - **Qmake**: simplifica el proceso de construcción de proyectos en las diferentes plataformas soportadas.
- ✓ Posee una GUI integrada y un diseñador de formularios.
- ✓ Ayuda integrada sensible al contexto.
- ✓ Depurador visual.
- ✓ Resaltado y auto-completado de código.

Capítulo 1: Fundamentación Teórica

1.5.7 Tecnología para la comunicación ZMQ 4.0.3

ZeroMQ es una tecnología que permite el intercambio de mensajes a través de comunicación TCP/IP de forma rápida y sencilla. Utiliza distintos patrones de distribución que permiten realizar balanceos de carga, redundancia de nodos y esquemas de servicios. Es una tecnología GPL y de código abierto, multiplataforma; utilizada en muchas empresas por la escalabilidad que proporciona (iMatix, 2014).

Para la solución propuesta se utilizará la tecnología de comunicación ZeroMQ, ya que es la determinada por el sistema Xilema Suria para establecer la comunicación entre sus módulos. Además, los sistemas de comunicación diseñados con esta biblioteca ofrecen rendimiento, robustez y compatibilidad con el lenguaje de programación seleccionado.

Conclusiones parciales

Con el análisis de los conceptos asociados al objeto de estudio, el campo de acción y las soluciones similares existentes, se obtuvo una visión de los elementos y características a tener en cuenta durante el diseño y construcción del multiplexor de flujos de video provenientes de cámaras IP. Se arribó a las siguientes conclusiones:

- ✓ La multiplexión juega un papel fundamental en la transmisión de los flujos de video dentro de los sistemas de video vigilancia, ya que su uso disminuye el tráfico de datos en la red y el volumen de procesamiento en el hardware de las estaciones de monitorización.
- ✓ El análisis de soluciones existentes tanto a nivel nacional como internacional, demostró que estas no satisfacen las necesidades de Xilema Suria debido a su alto costo o que no cumplen todos los requerimientos que necesita el sistema. Sin embargo, se obtuvieron de ellas características de funcionamiento que sirven de guía para el desarrollo de la investigación.
- ✓ El estudio de los procesos de captura y transmisión de los flujos de video provenientes de cámaras IP, garantizó un mejor entendimiento sobre el funcionamiento de los mismos en un sistema de video vigilancia.
- ✓ Las herramientas seleccionadas impulsan la soberanía tecnológica propuesta por el país. Las características y facilidades que ofrecen cada una de ellas, favorecen la integración del multiplexor con el sistema de video vigilancia Xilema Suria.

Capítulo 2: Análisis y Diseño del Sistema

Capítulo 2: Análisis y Diseño del Sistema

Introducción

En el presente capítulo se realiza un estudio del dominio del negocio, que tiene como fin comprender las características que posee el mismo y que luego serán implementadas en la solución. Se identifican los requisitos funcionales y no funcionales, agrupándose en casos de uso y presentando una descripción de cada uno de ellos. También se presentan los diagramas de clases del diseño y otros artefactos resultantes del flujo de trabajo de la metodología RUP, mostrándose un correcto uso de la arquitectura seleccionada y los patrones de diseño.

2.1 Modelo de dominio

Al no existir un negocio real, no es posible precisar la estructura de los procesos de negocio que intervienen en el dominio del problema, por ello se realizó un modelo de dominio. El modelo de dominio permite de manera visual mostrar al usuario los principales conceptos asociados a la situación problemática planteada. Es una representación de las clases conceptuales del mundo real, no de componentes de software y emplea un glosario de términos para lograr una mejor representación de los conceptos asociados (Larman, 2003).

2.1.1 Descripción del flujo del diagrama del modelo de dominio

El proceso de monitorización es iniciado por un usuario que utiliza el módulo Visor que se encuentra instalado en una estación de monitorización. Esta debe estar conectada a una red local para que pueda llevarse a cabo el proceso de visualización de los flujos de video. Desde este módulo se pueden visualizar y operar los flujos de video emitidos por una o varias cámaras. Cada cámara puede emitir uno o muchos flujos independientes, según las peticiones que se realicen del dispositivo. El módulo Visor interactúa con el módulo Gestor, que es el encargado de orquestar los procesos del sistema Xilema Suria.

Capítulo 2: Análisis y Diseño del Sistema

2.1.2 Diagrama de Modelo de dominio

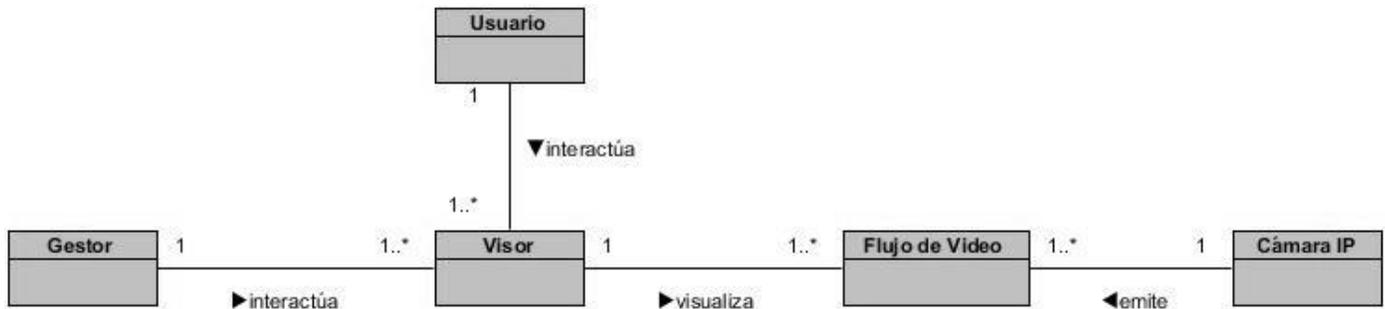


Fig. 1: Modelo de dominio multiplexor de flujos de video provenientes de cámaras IP para el sistema de video vigilancia Xilema Suria.

2.1.3 Descripción de las clases

Gestor: módulo fundamental del sistema Xilema Suria, encargado de la interacción entre los demás módulos del sistema. Controla y supervisa todo el tráfico de información.

Usuario: persona encargada de operar con el módulo Visor.

Visor: módulo del sistema Xilema Suria que permite la visualización de los flujos de video emitidos por las cámaras IP y además interactúa con el módulo Gestor. Es el módulo con el que se relacionan los usuarios. Tiene la capacidad de reflejar todo el aspecto organizativo con que se manejan las cámaras internamente, además de poder visualizarlas de manera independiente o colectiva (a manera de vistas). También permite manipularlas de acuerdo a las capacidades de cada una.

Flujos de Video: representan la secuencia de imágenes que transmiten las cámaras IP. Se pueden visualizar desde el módulo Visor.

Cámaras IP: videocámaras diseñadas para enviar los flujos de video obtenidos de un área en específico a través de la red.

2.2 Especificación de los requisitos de software

La especificación de requisitos software no es más que la descripción completa del comportamiento del sistema que se desea desarrollar. La misma incluye los requisitos funcionales, que son capacidades o condiciones que el sistema debe cumplir, y los requisitos no funcionales o complementarios, que son propiedades o cualidades que el producto debe tener.

Capítulo 2: Análisis y Diseño del Sistema

2.2.1 Requisitos Funcionales

Los requisitos funcionales son declaraciones de servicios que el sistema debe proporcionar, definen la manera en que debe reaccionar a determinadas entradas y cómo se debe comportar en situaciones particulares (Pressman, 2010). Estos requisitos son las características fundamentales del sistema y expresan la capacidad de acción del mismo.

Nota: Se usa el prefijo RF en su nomenclatura.

RF1: Capturar flujo de video

Descripción: El sistema debe permitir capturar el flujo de video proveniente de una cámara IP.

Entrada: Dirección URL de la cámara IP (Cadena de caracteres). Según el tipo de cámara, las direcciones URL tendrán la siguiente sintaxis:

http://ip:puerto/URLdecomandos?variable=valor

rtsp://usuario:contraseña@ip/URLdecomandos

Salida: Flujo de video emitido por la cámara IP.

RF2: Multiplexar video

Descripción: El sistema debe permitir multiplexar los flujos de video emitidos por un grupo de cámaras IP.

Entrada: Flujos de video emitidos por las cámaras y la disposición en que se quieren mostrar según la petición del módulo Visor.

Salida: Flujo de video multiplexado.

RF3: Transmitir flujo de video multiplexado

Descripción: El sistema debe transmitir el flujo de video multiplexado hacia la estación de monitorización que realizó la petición.

Entrada: Flujo de video multiplexado que se desea transmitir.

Salida: Transmisión de flujo de video.

RF4: Detener multiplexión de video

Descripción: El sistema debe permitirle al usuario detener una transmisión en el momento que lo desee.

Capítulo 2: Análisis y Diseño del Sistema

Entrada: Identificador de la transmisión multiplexada que se desea detener (Cadena de caracteres).

Salida: Fin de la transmisión del flujo de video multiplexado.

2.2.2 Requisitos No Funcionales

Los requisitos no funcionales son las restricciones de los servicios o funciones ofrecidas por el sistema, restringen el espacio de posibles soluciones. Debe pensarse en estos, como las características que hacen al producto atractivo, usable, rápido o confiable (Pressman, 2010).

Nota: Se usa el prefijo RnF en su nomenclatura.

Usabilidad

RnF1: Tipo de Aplicación Informática: El multiplexor de flujos de video provenientes de cámaras IP será una aplicación de escritorio.

RnF2: Finalidad: Proveer al proyecto de video vigilancia Xilema Suria de un multiplexor capaz de unir en un solo video los flujos provenientes de las cámaras IP conectadas al sistema, provocando la disminución del tráfico de red y del volumen de procesamiento de las estaciones de monitorización.

RnF3: Ambiente: Para su funcionamiento, el sistema requiere la instalación de cámaras IP que proveerán los flujos de video.

RnF4: Requisitos de Software

Biblioteca *GStreamer* 1.0.

Biblioteca ZMQ 4.0.3.

Sistema Operativo: Linux, distribución de Ubuntu 14.04.

RnF5: Requisitos de Hardware

El servidor donde se debe instalar el sistema debe tener altas prestaciones de hardware, ya que en el mismo se realiza un elevado volumen de procesamiento de video. Para el funcionamiento del sistema se debe disponer como mínimo del siguiente equipamiento:

Procesador Core i3-2120 a 3.30 GHz.

Memoria RAM de 2 GB.

Disco Duro de 10 GB.

Capítulo 2: Análisis y Diseño del Sistema

Dispositivo de red con una velocidad de 100 Mbps.

Confiabilidad

RnF6: Si se interrumpe la energía o la red en la estación de procesamiento donde se está ejecutando el multiplexor de video IP, el sistema Xilema Suria no debe obtener el flujo de video solicitado. Una vez reanudada la energía en la estación debe ejecutarse manualmente el sistema para que inicie sus funcionalidades.

Eficiencia

RnF7: Latencia¹⁵: la transmisión debe tener una latencia máxima de 20 segundos.

RnF8: Capacidad: El sistema debe permitir 10 conexiones simultáneas al multiplexor.

Restricciones del diseño

RnF9: El lenguaje de programación a utilizar en el desarrollo de la aplicación será C++, con el marco de trabajo Qt 5.3.

Restricciones de Integración

RnF10: Para realizar la integración entre el multiplexor de video IP y el sistema Xilema Suria se debe utilizar la biblioteca ZMQ en su versión 4.0.3.

2.3 Descripción del sistema.

Luego de identificados los requisitos funcionales del multiplexor de flujos de video provenientes de cámaras IP para el sistema Xilema Suria, se agruparon en casos de uso (CU), los cuales se representan en el diagrama de casos de uso del sistema. Cada uno de los casos de usos identificados engloba un conjunto de acciones, las cuales son inicializadas por los actores del sistema.

¹⁵ **Latencia:** tiempo que dura en llegar una acción desde su punto de inicio hasta su "punto de fuga", es decir cuando la acción se consume (QtGStreamer, 2014). En la presente investigación, es el tiempo que transcurre entre la captura del video por la cámara IP y su visualización en el sistema.

Capítulo 2: Análisis y Diseño del Sistema

2.3.1 Definición de los actores

Un actor del sistema no es más que un conjunto de roles que los usuarios desempeñan cuando interactúan con los casos de uso (Pressman, 2010). El actor identificado para el multiplexor de video IP es el Visor del sistema de video vigilancia Xilema Suria.

Tabla 1: Actores del sistema.

Actor	Descripción
Visor	Es el módulo del sistema de video vigilancia Xilema Suria que se comunica con el multiplexor de video IP a través del envío de peticiones TCP para ordenar la multiplexión de los flujos de video que se transmiten. También se encarga de detener una transmisión multiplexada.

2.3.2 Listado de los Casos de Uso.

Un CU es un fragmento de una funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores, especificando una secuencia de acciones o un determinado comportamiento que el sistema puede llevar a cabo durante la interacción con los actores, incluyendo alternativas dentro de la propia secuencia (Jacobson, y otros, 2000). De acuerdo a los RF definidos anteriormente, se agruparon en los siguientes casos de usos:

CU Transmitir video

RF1 Capturar flujo de video.

RF2 Multiplexar video.

RF3 Transmitir flujo de video multiplexado.

CU Detener multiplexor

RF4 Detener multiplexión de video.

2.3.3 Diagrama de Casos de Uso del Sistema

Un Diagrama de Casos de Uso del Sistema representa gráficamente a los procesos y sus interacciones con los actores (Jacobson, y otros, 2000). La siguiente figura muestra el diagrama de casos de uso del multiplexor de flujos de video provenientes de cámaras IP.

Capítulo 2: Análisis y Diseño del Sistema

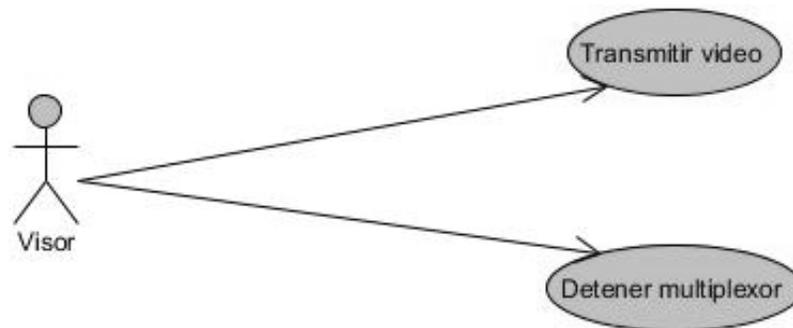


Fig. 2: Diagrama de CU multiplexor de flujos de video provenientes de cámaras IP.

Al recibir una solicitud de transmisión de varias cámaras IP, el Visor del sistema Xilema Suria se comunica con el multiplexor de video IP, el cual captura los flujos de video emitidos por las cámaras IP, los multiplexa y finalmente transmite el flujo de video multiplexado hacia las estaciones de monitorización. El actor Visor es también el encargado de detener las transmisiones de flujos de video multiplexados.

2.3.4 Especificación de los Casos de Uso del Sistema

En el presente epígrafe se presenta la descripción textual del CU Transmitir video. Las descripción del CU Detener multiplexor pueden encontrarse en el Anexo II del presente trabajo.

Tabla 2: Descripción del CU Transmitir video.

CU	Transmitir video.
Objetivo	El objetivo de este CU es permitir la captura, multiplexión y transmisión de los flujos de video hacia las estaciones de monitorización.
Actores	Visor.
Resumen	El caso de uso inicia cuando el Visor del sistema Xilema Suria se comunica con el multiplexor a través del envío de mensajes para solicitar la transmisión de los flujos de video emitidos por varias cámaras IP. Una vez hecha la petición se capturan los flujos emitidos por los dispositivos de seguridad y se multiplexan, enviándose un único video hacia las estaciones de monitorización. El CU termina cuando se cancelen las peticiones de transmisión.
Complejidad	Alta.
Prioridad	Crítica.
Precondiciones	Multiplexor de video IP iniciado. Debe existir conexión a las cámaras IP.

Capítulo 2: Análisis y Diseño del Sistema

Postcondiciones	Transmisión del flujo de video a través los protocolos UDP o RTSP hacia las estaciones de monitorización, por un puerto y dirección IP local de la computadora donde se está ejecutando el multiplexor.	
Flujo de eventos		
Flujo básico: Transmitir video.		
	Actor	Sistema
1.	Realiza una petición con el listado de cámaras que desea multiplexar y la posición y el tamaño que ocupan cada una de estas en el mosaico en el que se mostrarán, así como el protocolo que se desea utilizar (UDP o RTSP).	
2.		Recibe las direcciones URL de las cámaras IP.
3.		Solicita a la cámara IP el flujo de video.
4.		Crea un elemento con las posiciones suministradas por el usuario para mostrar las cámaras.
5.		Redimensiona el flujo de video para poder ubicarlo en la cuadrícula de visualización.
6.		Asigna cada flujo de video a la posición que le corresponde dentro de la cuadrícula.
7.		Genera la dirección del flujo de video a transmitir.
8.		Comienza a transmitir por los protocolos UDP o RTSP el flujo de video multiplexado.
9.		Envía al cliente la dirección y el identificador correspondiente a la transmisión. Termina el CU.
Flujos alternos.		
3a. Cámara IP apagada.		
	Actor	Sistema
3a.		La cámara IP se encuentra apagada y no devuelve flujo de video. En su lugar se mostrará una pantalla en negro. Continúa en el paso 4 del flujo básico.
Flujos alternos.		
3b. No se puede multiplexar video.		

Capítulo 2: Análisis y Diseño del Sistema

	Actor	Sistema
3b.		El sistema no recibe los flujos provenientes de las cámaras IP debido a que estas están apagadas. Al no recibir ningún video de las cámaras el sistema no inicia la multiplexión. Termina el CU.
Relaciones	CU Incluidos	No procede.
	CU Extendidos	No procede.
Referencias	RF 1, RF 2, RF3, RnF 3, RnF 5, RnF 6, RnF 7, RnF 8, RnF 9, RnF 10.	

2.4 Descripción de la Arquitectura

De acuerdo al *Software Engineering Institute* (SEI), la Arquitectura de Software se refiere a “*la organización fundamental de un sistema encarnada en sus componentes, las relaciones de los componentes con cada uno de los otros y con el entorno, y con los principios que orientan su diseño y evolución*” (Bass, y otros, 2003). La manera en que se organiza un sistema tiene un impacto directo sobre la capacidad de este para satisfacer los atributos de calidad del sistema, los que forman parte de los requisitos no funcionales del sistema.

Los estilos de llamada y retorno se enfatizan en la modificabilidad y la escalabilidad. Son los estilos más generalizados en sistemas de gran escala. Miembros de esta familia son las arquitecturas de programa principal y subrutina, los sistemas basados en llamadas a procedimientos remotos, los sistemas orientados a objetos y los sistemas jerárquicos en capas (Reynoso, y otros, 2004). Los patrones arquitectónicos definidos para el multiplexor de video IP pertenecen a este estilo.

2.4.1 Arquitectura Cliente Servidor

Se realizó un estudio de las diferentes arquitecturas de software, determinándose la Cliente Servidor(C/S) como la más adecuada para el multiplexor de video IP. Esta arquitectura es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. El cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio) (Leal, y otros, 2011). Se escoge esta arquitectura ya que el sistema que se desea desarrollar deberá satisfacer las peticiones provenientes desde uno o varios visores del sistema de video vigilancia Xilema Suria. Cuando cada instancia del módulo Visor realiza peticiones de transmisión a las cámaras IP, envía un mensaje al multiplexor de video IP que se encarga de capturar los flujos de video de todos los

Capítulo 2: Análisis y Diseño del Sistema

dispositivos que el sistema necesita visualizar y los envía a la estación de monitorización en un único flujo de salida.

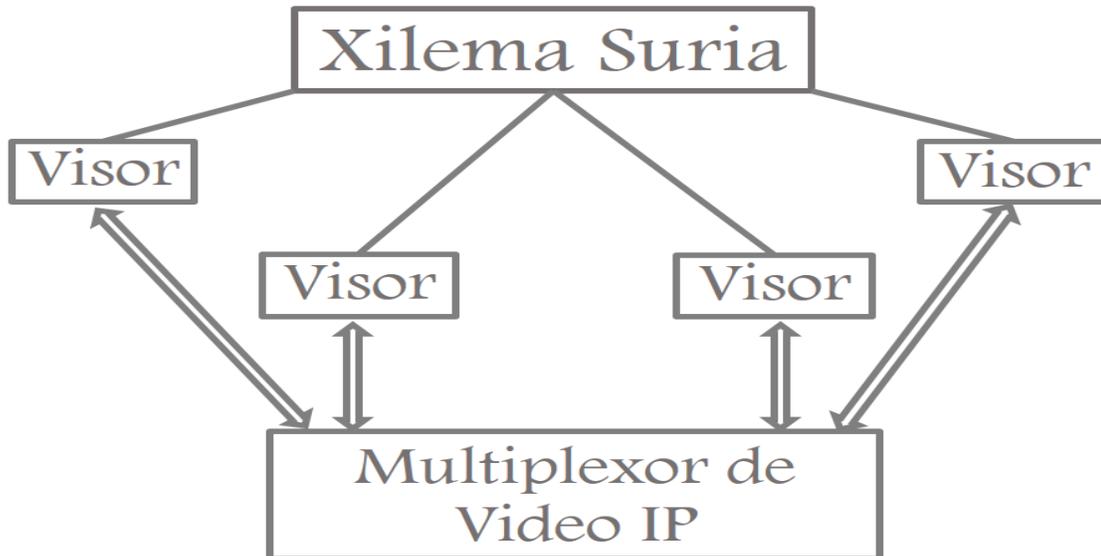


Fig. 3: Representación de la Arquitectura Cliente Servidor.

En la arquitectura C/S el **cliente** (Visor):

- Es quien inicia solicitudes o peticiones, tiene un papel activo en la comunicación.
- Espera y recibe las respuestas del servidor.
- Puede conectarse a varios multiplexores a la vez.

Por otra parte el **servidor** (multiplexor de video IP):

- Al iniciarse espera a que lleguen las solicitudes de los clientes, desempeña entonces un papel pasivo en la comunicación.
- Tras la recepción de una solicitud, la procesa y luego envía la respuesta al cliente.
- Acepta las conexiones de múltiples clientes.

La elección de la arquitectura C/S se debió a las siguientes **características** (Reynoso, y otros, 2004):

- Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes, distribuidas a lo largo de la red.

Capítulo 2: Análisis y Diseño del Sistema

- Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad y capacidades del disco y dispositivos de entrada/salida.
- El ambiente es heterogéneo. La plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre los mismos. Precisamente una de las principales ventajas de esta arquitectura es la posibilidad de conectar clientes y servidores independientemente de sus plataformas.

Entre las **ventajas** de usar la arquitectura C/S se encuentran (Leal, y otros, 2011):

- **Escalabilidad:** se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- **Fácil mantenimiento:** al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor sin sufrir muchas afectaciones.
- **Abundantes tecnologías:** existen técnicas suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la amigabilidad de la interfaz, y la facilidad de empleo.

2.4.2 Arquitectura en Capas

Es una arquitectura organizada jerárquicamente en capas, donde cada capa provee servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inferior. La interacción entre las capas está limitada solo a las que son adyacentes. Al dividir un sistema en N-capas, cada capa puede tratarse de forma independiente, sin tener que conocer los detalles de las demás (Reynoso, y otros, 2004).

La división de un sistema en capas facilita el diseño modular, permite la construcción de sistemas débilmente acoplados, lo que significa que si se disminuye las dependencias entre capas, resulta más fácil modificar la implementación de una capa sin afectar al resto del sistema. Este patrón permite descomponer el problema en una secuencia de pasos incrementales, basado en niveles de abstracción (Reynoso, y otros, 2004).

Capítulo 2: Análisis y Diseño del Sistema

Se escogió este tipo de arquitectura para el desarrollo de la aplicación, pues con la misma se reducen las dependencias existentes entre las clases, de forma tal que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Además, cada nivel posee funcionalidades diferentes, lo que permite el diseño de una arquitectura escalable que puede ser ampliada en caso de ser necesario.

Arquitectura en Dos Capas

Para el desarrollo del multiplexor de video IP se definió una arquitectura en dos capas como una variante del patrón N-capas. Las mismas son denominadas como Capa de Comunicación y Capa de Negocio.

Capa de Comunicación: Es la encargada de establecer la comunicación entre el módulo Visor de Xilema Suria y el multiplexor de video IP. Recibe las peticiones de transmisión y las envía a la Capa de Negocio. Al estar listo el proceso de multiplexión, envía al Visor el identificador de la transmisión iniciada.

Capa de Negocio: En esta capa se establecen todas las reglas que deben cumplirse, es donde se gestiona y se realizan las operaciones solicitadas por el usuario. En la investigación, es la capa que se encarga de la captura, multiplexión y transmisión de los flujos de video. Se relaciona con la Capa de Comunicación, para recibir las solicitudes que envía el módulo Visor y presentar el identificador de la transmisión iniciada.

2.5 Patrones de diseño

Los patrones de diseño, constituyen soluciones estándar que brindan respuesta a un problema común en materia de diseño de sistemas e implementación. Estos mejoran la comunicación entre los diseñadores, facilitando además el aprendizaje del programador inexperto y logran crear parejas problema-solución. Cada patrón sugiere además numerosas ventajas, dentro de ellas la reutilización de código y el permitir la realización de un diseño apto para el cambio. Los patrones de diseño se aplican a un elemento específico de diseño como un agregado de componentes, para resolver algún problema de diseño, relaciones entre los componentes o los mecanismos para efectuar las comunicaciones de componente a componente (Gamma, y otros, 2004).

Los Patrones Generales de Software para Asignar Responsabilidades o *General Responsibility Assignment Software Patterns* (**GRASP**) describen los principios fundamentales del diseño de objetos para la asignación de responsabilidades (Gamma, y otros, 2004). Teniendo en cuenta el patrón

Capítulo 2: Análisis y Diseño del Sistema

arquitectónico seleccionado, los patrones de diseño que se aplican antes de la implementación del multiplexor son los siguientes:

- ✓ **Experto:** Es el principio básico de la asignación de responsabilidades. Este patrón garantiza que cada clase cumpla con sus responsabilidades, según la información que contenga y las funcionalidades que se deseen implementar (Larman, 2000). En la aplicación cada clase mantiene este principio, por ejemplo la clase *Transmitter*, es la experta en conocer todo lo relacionado al video en transmisión (*streaming*), controlando las acciones relacionadas al procesamiento del flujo de video que se transmite.
- ✓ **Alta Cohesión:** Es la idea de tener las clases lo menos ligadas entre sí. De este modo, si se produce una modificación en alguna de ellas, ocasionará poca repercusión en el resto de las clases, fortaleciendo la reutilización, y disminuyendo la dependencia entre clases (Larman, 2000). Se utilizó Alta Cohesión para asignar responsabilidades a clases que estén altamente relacionadas con el procesamiento y la transmisión de video, garantizando la contribución entre estas para realizar tareas de elevada complejidad en vez de utilizar una sola clase para ello, por ejemplo, *VideoStream*, *VideoFilter*, *VideoMixer* y *Transmitter*.
- ✓ **Controlador:** Permite asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas (Larman, 2000). Se utilizó para decidir qué clase es la responsable de recibir o manejar un evento del sistema, por ejemplo la clase *ControllerClass* es la controladora, pues es la encargada de recibir los datos de cada solicitud de transmisión y enviarlos a las distintas clases para su procesamiento.
- ✓ **Bajo Acoplamiento:** Es la medida de qué tan fuertemente está conectada una clase con las demás, es decir, cuántas clases conoce y necesita. Una clase con bajo acoplamiento no depende de muchas otras clases. Tener un bajo acoplamiento, soporta el diseño de clases más independientes, que reducen el impacto de los cambios y permite una mayor reutilización de código (Larman, 2000). Este patrón fue utilizado en todas las clases de la solución propuesta.
- ✓ **Creador:** El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos. Se aplica en todos los casos donde una clase tiene la responsabilidad de crear una nueva instancia de la otra. Se utilizó para decidir qué clase es la responsable de la construcción de objetos,

Capítulo 2: Análisis y Diseño del Sistema

es decir, la que le puede asignar los datos al constructor, por ejemplo las clases *ControllerClass*, *VideoFilter*, *VideoMixer* y *Transmitter*.

Los patrones creados por la Banda de los Cuatro o *Gang of Four (GoF)*¹⁶, son agrupados en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento (Gamma, y otros, 2004). Los estructurales tratan la combinación de clases, su relación y la formación de estructuras de alta complejidad, mientras que los creacionales tratan la creación de instancias y los de comportamientos tratan la interacción y la cooperación entre clases. En el desarrollo de la solución propuesta se hace uso del patrón Observador, perteneciente a este último grupo.

- ✓ **Observador:** Define una dependencia “uno a muchos” entre objetos, para que cuando uno de ellos cambie su estado, todos los que dependan de él sean avisados y puedan actualizarse convenientemente (Gamma, y otros, 2004). Este patrón se evidencia en las clases *CommunicationServer* y *VideoMultiplexerManager* para notificar que un cliente ha hecho una nueva petición para iniciar o detener una transmisión.

En el desarrollo del multiplexor de video IP se hace uso de otros patrones de diseño como:

- ✓ **Layer Supertype** (Capa Supertipo): Este patrón proporciona una clase base que contiene características comunes para otras clases dentro de la misma capa. Su uso evita que clases de una misma capa tengan métodos duplicados (Fowler, y otros, 2002). En el multiplexor de video IP las clases *VideoStream*, *VideoFilter*, *VideoMixer* y *Transmitter* hacen uso de funcionalidades comunes implementadas en la clase *MyGstElement* que actúa como una superclase base para ellas.
- ✓ **Singleton Factory** (Fábrica Única): permite tener un único objeto compartido entre muchos hilos para el intercambio de datos entre instancias (Fowler, y otros, 2002). La clase *ControllerClass* asegura que se devuelva un sólo objeto de la clase *VideoStream* por cada cámara solicitada.

2.6 Modelo del diseño

Un modelo de diseño es un modelo de objetos físicos que describe la realización física de los casos usos. Se centra en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas

¹⁶ **GoF:** Nombre con el que se conoce comúnmente a los autores del libro *Design Patterns* (ISBN 0-201-63361-2), referencia en el campo del diseño orientado a objetos. La Banda de los Cuatro la componen los autores Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides.

Capítulo 2: Análisis y Diseño del Sistema

con el entorno de implementación, tienen impacto en el sistema. El diseño representa el centro de atención al final de la fase de elaboración y es el comienzo de las iteraciones de construcción. Este representa un plano del modelo de implementación, garantizando de esta manera una arquitectura sólida y estable. El modelo de diseño perdura durante todo el ciclo de vida del software y constituye la entrada fundamental utilizada para el correcto desarrollo de la implementación (Jacobson, y otros, 2000).

2.6.1 Diagrama de secuencia del diseño

Un diagrama de secuencia es una forma de diagrama de interacción, que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo, representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino. Los diagramas de secuencia no están pensados para mostrar lógicas de procedimientos complejos (Jacobson, y otros, 2000). A continuación se muestra el diagrama de secuencia para el CU Transmitir video. El diagrama correspondiente al CU Detener multiplexor puede encontrarse en el Anexo III del presente trabajo.

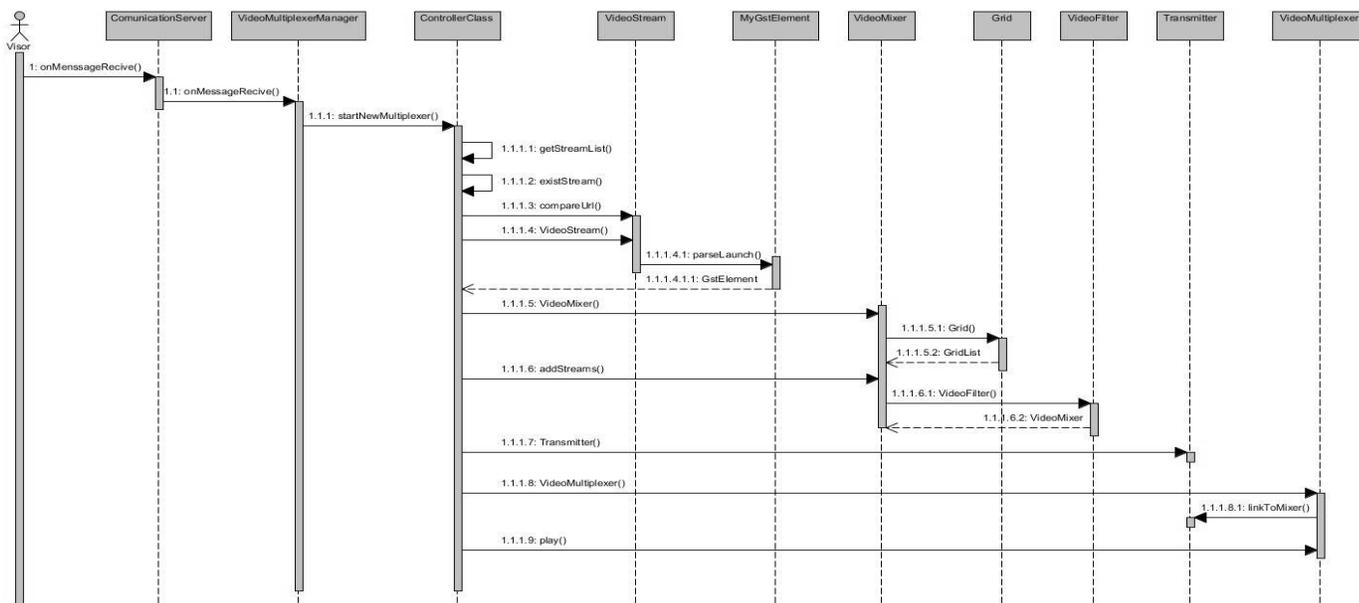


Fig. 4: Diagrama de Secuencia CU Transmitir video.

2.6.2 Clase del diseño

Una clase del diseño es una abstracción de una clase en la implementación del sistema. El lenguaje utilizado para especificar una clase del diseño es el mismo que el lenguaje de programación. Esto conlleva

Capítulo 2: Análisis y Diseño del Sistema

a que las operaciones, atributos y tipos son especificados utilizando la sintaxis del lenguaje de programación elegido (Larman, 2000).

2.6.3 Diagrama de Clases del Diseño

Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema. Esto incluye modelar el vocabulario del sistema, modelar las colaboraciones o modelar esquemas. Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales; sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa (Jacobson, y otros, 2005). En la siguiente ilustración, se puede observar una representación del diagrama de clases del diseño del multiplexor de video IP.

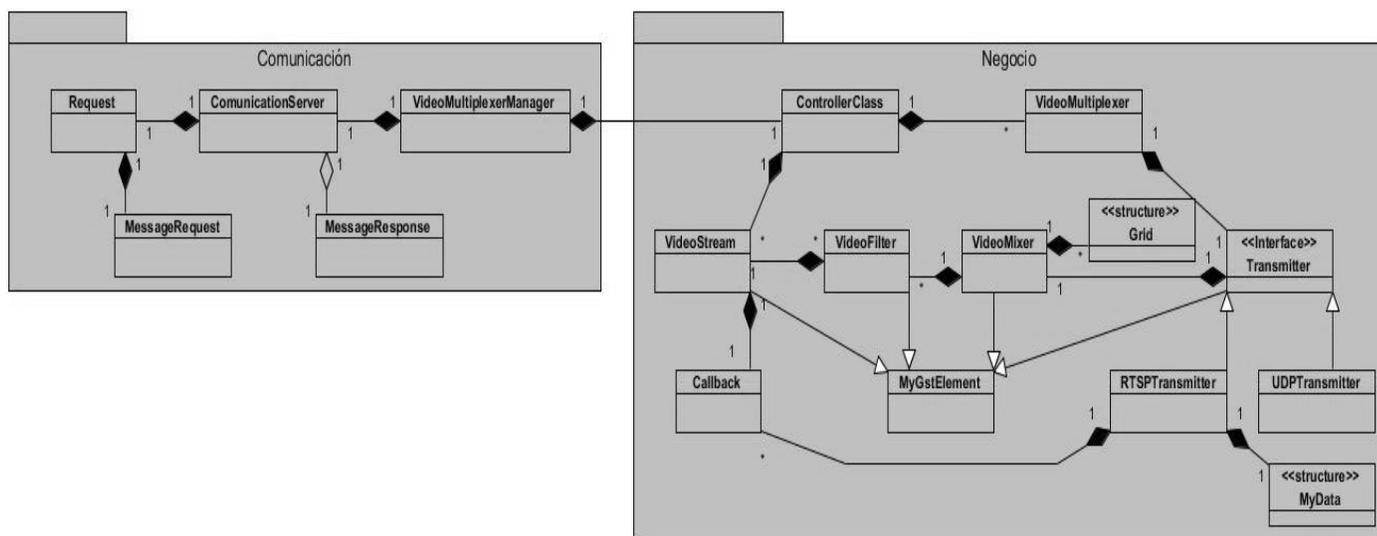


Fig. 5: Representación del Diagrama de Clases del Diseño.

La clase *ControllerClass* es la clase controladora, se encarga de comenzar a transmitir una nueva multiplexión o de detener una que ya esté emitiéndose. Posee una lista de objetos de tipo *VideoStream* que le permite controlar las conexiones a las cámaras IP que están transmitiendo. También contiene una lista de objetos *VideoMultiplexer* para controlar los multiplexores que están transmitiendo en ese momento. La clase *VideoFilter* funciona como un filtro que redimensiona el video antes de ingresarlo a la clase *VideoMixer* la cual crea una rejilla y ubica los *streaming* de videos filtrados en una disposición

Capítulo 2: Análisis y Diseño del Sistema

predefinida por el usuario. Dicha disposición se obtiene a través de la estructura *Grid* que contiene las coordenadas donde será ubicado cada flujo de video, así como el tamaño del mismo.

La interfaz *Transmitter* proporciona métodos virtuales que necesitan implementarse en las clases *UDPTransmitter* y *RTSPTransmitter*. Estas clases se encargan de transmitir por los protocolos UDP o RTSP según la petición realizada. La clase *Callback* permite a las clases *RTSPTransmitter* y *VideoStream* compartir un mismo flujo entre varias peticiones. *MyGstElement* es la clase que maneja las funcionalidades y procedimientos nativos de la biblioteca *GStreamer* que son comunes para las clases *VideoStream*, *VideoFilter*, *VideoMixer* y *Transmitter*.

En la Capa de Comunicación, la clase *CommunicationServer* es la encargada de atender las peticiones del cliente. Accede a las funcionalidades de la biblioteca ZMQ para iniciar y finalizar cada transmisión. La clase *VideoMultiplexerManager* es la encargada de comunicarse con la Capa de Negocio, específicamente con la clase *ControllerClass*, enviándole los datos de la solicitud y recibiendo de vuelta los datos de la transmisión. Las clases *Request*, *MessageRequest* y *MessageResponse* son tipos de mensajes utilizados por *CommunicationServer* para establecer la comunicación.

Conclusiones parciales

En este capítulo se definieron temas relacionados con el análisis y el diseño del multiplexor de flujos de video provenientes de Cámaras IP para el sistema Xilema Suria, arribándose a las siguientes conclusiones:

- ✓ La selección de una arquitectura en capas para el multiplexor de video IP facilita la reutilización de las capas, la estandarización y la contención de cambios, proporcionando mejor desempeño, robustez, portabilidad, flexibilidad y escalabilidad a la aplicación.
- ✓ Los patrones de diseño aplicados ayudaron a complementar la arquitectura del sistema, evidenciando el uso de soluciones demostradas como parte de las buenas prácticas de la programación.
- ✓ Los diagramas de secuencias del diseño permiten una mejor comprensión del flujo de vida en cada caso de uso. Con el diagrama de clases de diseño, se obtuvo una visión estática del sistema, facilitando la comunicación entre los programadores y el descubrimiento de fallas de la aplicación en el diseño.

Capítulo 3: Implementación y Prueba

Capítulo 3: Implementación y Prueba

Introducción

En este capítulo se presentan todos los elementos relacionados al flujo de trabajo de implementación, describiéndose las técnicas utilizadas en el desarrollo de la aplicación. Se realizan los diagramas de componentes de implementación y de despliegue resultantes de esta etapa de construcción. Se describen las pruebas a las que fue sometida la aplicación, con el objetivo de validar su correcto funcionamiento.

3.1 Modelo de Implementación

El modelo de implementación está conformado por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes podemos encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos (Sommerville, 2005). El modelo de implementación muestra:

- ✓ Las dependencias entre las partes de código del sistema (diagramas de componentes).
- ✓ La estructura del sistema en ejecución (diagrama de despliegue).

3.1.1 Diagrama de Despliegue

El Diagrama de Despliegue muestra las relaciones físicas de los distintos nodos que componen un sistema y las relaciones existentes entre ellos (Larman, 2000). Al graficarse el entorno físico en que se desarrolla el proceso de video vigilancia, se pueden comprender cuestiones claves como la distribución del sistema, que influyen directamente en la implementación.

A continuación se describen los nodos que integran el diagrama de despliegue.

Visor: módulo del sistema de video vigilancia Xilema Suria que permite la visualización de los flujos de video emitidos por las cámaras IP.

Multiplexor de video IP: aplicación que recibe los flujos de video provenientes de las cámaras IP, los integra en un solo video y los transmite.

Cámara: dispositivo encargado de emitir los flujos de video.

Capítulo 3: Implementación y Prueba

A través del módulo *Visor* del sistema Xilema Suria, el usuario realiza una petición de transmisión de varias *cámaras IP* conectadas al sistema. Estas solicitudes (realizadas mediante el protocolo TCP) son recibidas por el *Multiplexor de video IP* que emplea los protocolos y HTTP para comunicarse con las cámaras y transmite el flujo de video multiplexado hacia el módulo *Visor* utilizando los protocolos UDP o RTSP.

En la siguiente figura se representa el diagrama de despliegue correspondiente a la aplicación:

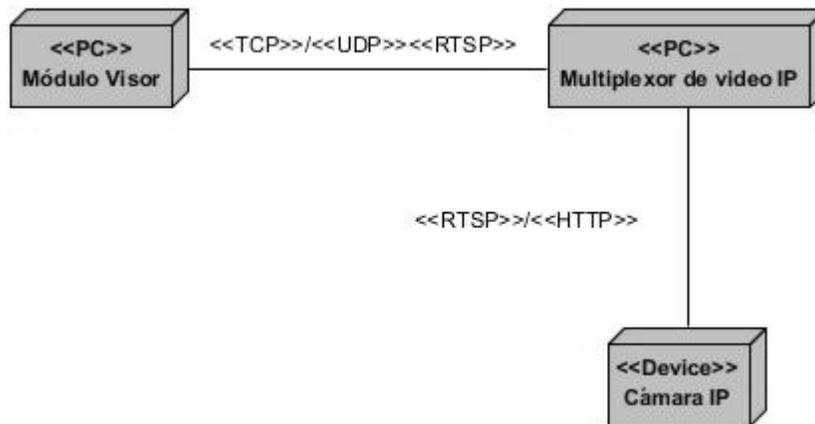


Fig. 6: Diagrama de Despliegue.

3.1.2 Diagrama de componentes de implementación

El diagrama de componentes de implementación permite describir los elementos físicos que componen el sistema y las relaciones entre ellos. Los componentes representan todos los tipos de elementos de software que intervienen en la fabricación de aplicaciones informáticas. Estos pueden ser simples archivos, paquetes o bibliotecas cargadas dinámicamente (Jacobson, y otros, 2000). En la siguiente figura se muestra el diagrama de componentes de implementación correspondiente a la aplicación:

Capítulo 3: Implementación y Prueba

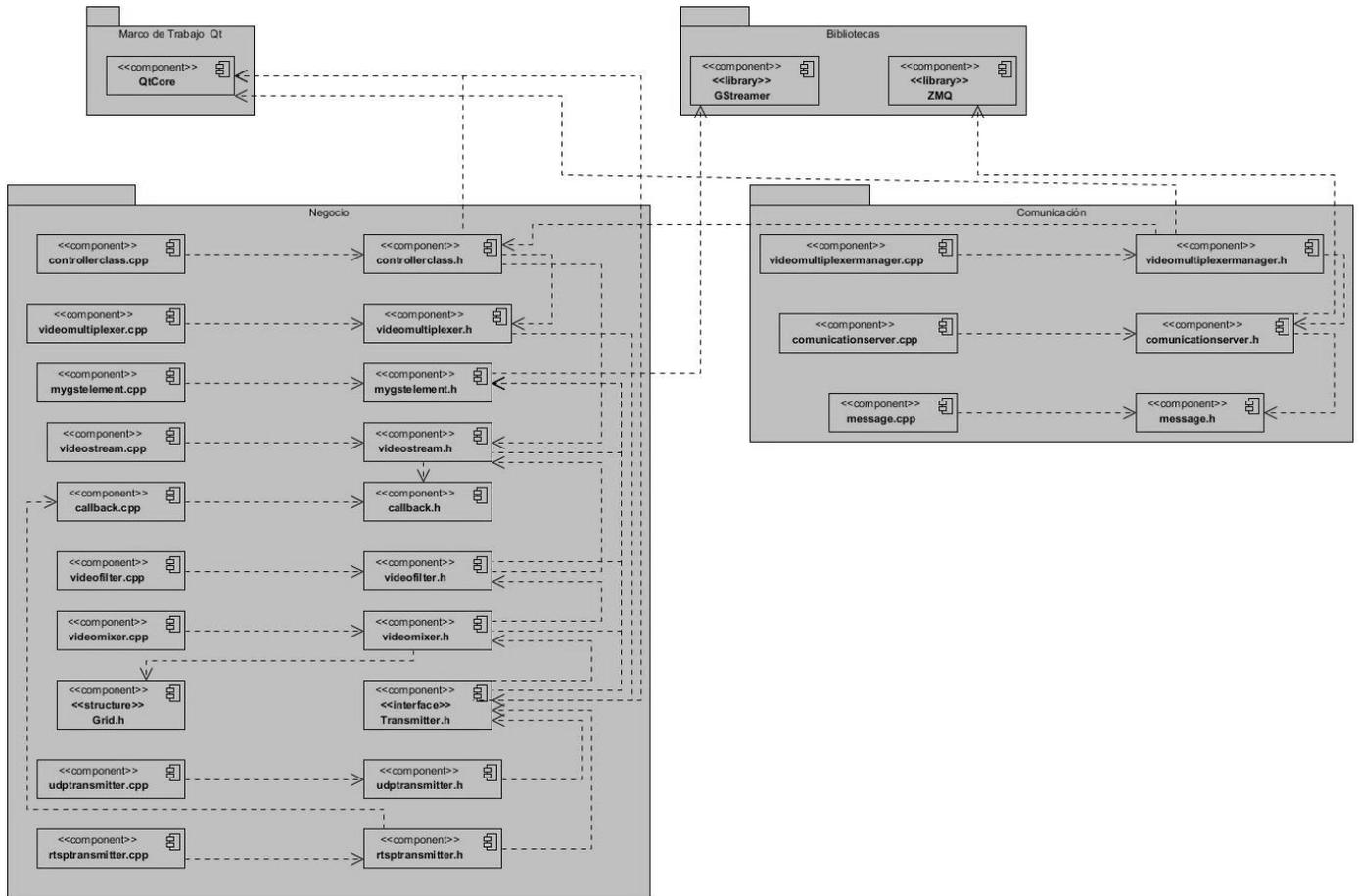


Fig. 7: Diagrama de Componentes de Implementación.

3.2 Pruebas de Software

En un proyecto de desarrollo de software los errores pueden presentarse en cualquiera de las etapas del ciclo de vida del software. Aunque se intente detectarlos después de cada fase utilizando técnicas como la inspección, algunos errores permanecen sin ser descubiertos. Las pruebas de software no son más que técnicas de validación para verificar el resultado de la implementación probando cada construcción. Las mismas tienen como objetivo probar el funcionamiento del software en su máxima capacidad y tratar de corregir todos los errores existentes en el mismo (Pressman, 2010). Son un elemento crítico para la garantía de la calidad del software y representan una revisión final de las especificaciones del diseño y la codificación.

Capítulo 3: Implementación y Prueba

3.2.1 Prueba de Integración

Incluso cuando los módulos de un programa funcionan bien por separado es necesario probarlos conjuntamente. Un módulo puede tener un efecto adverso o inadvertido sobre otro módulo, por ejemplo, las subfunciones, cuando se combinan, pueden no producir la función principal deseada; la imprecisión aceptada individualmente puede crecer hasta niveles inaceptables al combinar los módulos; los datos pueden perderse o malinterpretarse entre interfaces (Juristo, y otros, 2006). Por lo tanto, es necesario probar el software ensamblando todos los módulos y probando que los componentes de la aplicación funcionan correctamente actuando en conjunto. Con este objetivo se crearon las pruebas de integración.

A menudo hay una tendencia a intentar una integración no incremental; es decir, a combinar todos los módulos y probar todo el programa en su conjunto. El resultado puede ser un poco caótico con un gran conjunto de fallos y la consiguiente dificultad para identificar el módulo (o módulos) que los provocó.

Otra variante es aplicar la integración incremental, en la que el programa se prueba en pequeñas porciones donde los fallos son más fáciles de detectar (Juristo, y otros, 2006). Existen dos tipos de integración incremental, la denominada ascendente (*down-top*) y descendente (*top-down*), siendo la segunda la utilizada en la presente investigación ya que es la recomendada para integrar módulos generales. La técnica *top-down* comienza con los módulos de nivel superior, y se verifica que los módulos de nivel superior llaman a los de nivel inferior de manera correcta, con los parámetros correctos. Los módulos subordinados al módulo de control principal se van incorporando a la estructura, bien de forma primero-en-profundidad, o bien de forma primero-en-anchura. La integración primero-en-profundidad integra todos los módulos de un camino de control principal de la estructura.

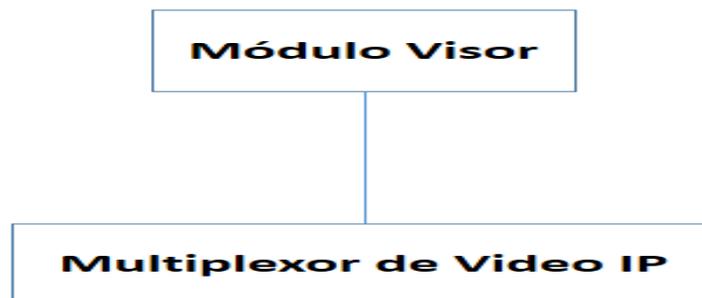


Fig. 8: Representación de la Integración Incremental Descendente.

La comunicación entre el multiplexor de video IP y el módulo Visor de Xilema Suria se logra haciendo uso de la biblioteca ZMQ que es la establecida por el sistema de video vigilancia para realizar la comunicación

Capítulo 3: Implementación y Prueba

entre sus módulos. En esta investigación, la comunicación se realizó a través de una aplicación de prueba, ya que no se ha desarrollado en el módulo Visor la interfaz necesaria para la comunicación con el multiplexor de video IP. La ejecución de la prueba queda explicada en los siguientes pasos:

1. La aplicación cliente envía un mensaje donde van contenidas las direcciones URL de las cámaras IP, la estructura de visualización, así como la dirección IP y el puerto al que se va a transmitir.
2. El multiplexor recibe el mensaje con los datos y transmite hacia el módulo Visor un video multiplexado que contiene los flujos de video de las cámaras IP.
3. La aplicación cliente recibe correctamente el video multiplexado con los flujos de video provenientes de las cámaras IP y los datos asociados a la transmisión.

3.2.2 Prueba de Sistema

La prueba del sistema abarca una serie de pruebas diferentes cuyo propósito principal es ejercitar intensamente el sistema de cómputo. Aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se hayan integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas (Pressman, 2010). En esta etapa se realizan las pruebas de Carga, Estrés y Rendimiento que permiten validar y verificar otros atributos de la calidad del sistema, tales como, la escalabilidad, la fiabilidad y el uso de los recursos.

Los resultados obtenidos en las pruebas realizadas, corresponden a una estación de trabajo con las siguientes prestaciones:

Procesador Core i3-2120 a 3.30 GHz.

Memoria RAM de 2 GB.

Pruebas de Carga: estas pruebas permiten determinar y validar la respuesta de la aplicación cuando es sometida a una carga de usuarios y/o transacciones que se espera en el ambiente de producción (Lara García, 2013). En esta prueba se comprobó el desempeño del multiplexor de video IP ante múltiples peticiones de multiplexión de video de forma simultánea. También permitió determinar la latencia que presentan los videos transmitidos por el multiplexor. A continuación se presentan los resultados de la prueba de carga realizada.

Capítulo 3: Implementación y Prueba

Tabla 3: Resultados de la Prueba de Carga aplicada al multiplexor de video IP.

Cantidad de Transmisiones	Transmisión UDP			Transmisión RTSP			Respuesta del Sistema
	CPU (%)	RAM (%)	Latencia (s)	CPU (%)	RAM (%)	Latencia (s)	
1	12	4.4	2	17	15.4	4	Multiplexa y transmite correctamente el flujo de video de las cámaras IP solicitadas.
2	22	8.0	2	26	30.2	5	
4	31	12.9	3	37	40.7	7	
6	43	19.2	4	49	58.1	10	
8	53	26.8	7	60	72.0	12	
10	65	40.2	8	80	86.3	13	

Pruebas de Estrés: pruebas para encontrar el volumen de datos o de tiempo en que la aplicación comienza a fallar o es incapaz de responder a las peticiones. Son pruebas de carga o rendimiento, pero superando los límites esperados en el ambiente de producción y determinados en las pruebas (Lara García, 2013). La meta de las pruebas de estrés es identificar y documentar las condiciones bajo las cuales el sistema falla. La realización de esta prueba determinó que para los requisitos de hardware anteriormente expuestos, la aplicación puede transmitir correctamente hasta 10 peticiones de multiplexión simultáneas. A continuación se presentan los resultados de la prueba de estrés realizada, probando el funcionamiento del multiplexor con más de 10 transmisiones al mismo tiempo.

Tabla 4: Resultados de la Prueba de Estrés aplicada al multiplexor de video IP.

CT ¹⁷	Transmisión UDP			Transmisión RTSP			Respuesta del Sistema
	CPU (%)	RAM (%)	Latencia (s)	CPU (%)	RAM (%)	Latencia (s)	
12	75	50.4	12	88	86.3	17	El sistema responde todas las peticiones pero no establece la conexión con algunas de las cámaras.
14	90	80	18	98	94	25	El sistema responde solo algunas peticiones.
16	-	-	-	-	-	-	El sistema colapsa.

Pruebas de rendimiento: estas pruebas se realizan para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo. Permiten probar cuán rápida y eficiente puede ser la

¹⁷ CT: Cantidad de transmisiones.

Capítulo 3: Implementación y Prueba

respuesta del sistema en un entorno controlado, que debe simular lo más cercano a la realidad, el entorno de producción donde se utilizará el software. También pueden servir para investigar, medir y validar el uso de los recursos que conlleva la ejecución del software ante volúmenes de carga esperados (Lara García, 2013).

En esta prueba se comprobó los recursos de RAM, CPU y red que consume el módulo Visor del sistema Xilema Suria para visualizar cantidades particulares de cámaras IP y se comparó con el gasto de recursos que conlleva este mismo proceso con el empleo del multiplexor de video IP. Los resultados obtenidos demuestran que el uso del multiplexor posibilita un mejor uso del ancho de banda disponible. Además, disminuye el impacto del sistema de video vigilancia sobre el hardware de las estaciones de monitorización.

Para medir estos recursos se usó el Administrador de Tareas de Windows 8.1¹⁸, herramienta incorporada a los sistemas operativos de Microsoft para obtener información de los procesos que se ejecutan en el equipo, además de supervisar el rendimiento del mismo. Se seleccionó esta herramienta ya que la versión actual del sistema Xilema Suria solo funciona en el sistema operativo Windows. La siguiente tabla muestra los valores de RAM (Megabits), CPU (%) y red (Megabits/segundo) que consume el módulo Visor del sistema Xilema Suria para visualizar los flujos emitidos por las cámaras IP, con y sin el uso del multiplexor de video IP.

Tabla 5: Resultados de las Pruebas de Rendimiento aplicadas al módulo Visor de Xilema Suria.

Cantidad de Cámaras	Módulo Visor sin usar Multiplexor			Módulo Visor usando Multiplexor		
	RAM (MB)	CPU (%)	Red (Mbps)	RAM (MB)	CPU (%)	Red (Mbps)
5	102.9	5.2	1.8	50.4	1.8	0.9
10	160.3	7.8	3.2	81.3	3.5	1.0
15	346.7	20	5.1	109.2	4.2	1.2
20	621.5	48.7	7.2	118.6	4.9	1.4
25	693.5	63.5	12.4	140.2	5.3	1.8

A continuación se representan de manera gráfica, los resultados obtenidos en las Pruebas de Rendimiento aplicadas al módulo Visor del sistema Xilema Suria.

¹⁸ **Administrador de Tareas:** En la opción **Procesos**, se observan datos de las aplicaciones que se encuentran en ejecución, como por ejemplo, su nombre, estado y porcentajes de utilización de la CPU, la memoria, el disco duro y la red.

Capítulo 3: Implementación y Prueba

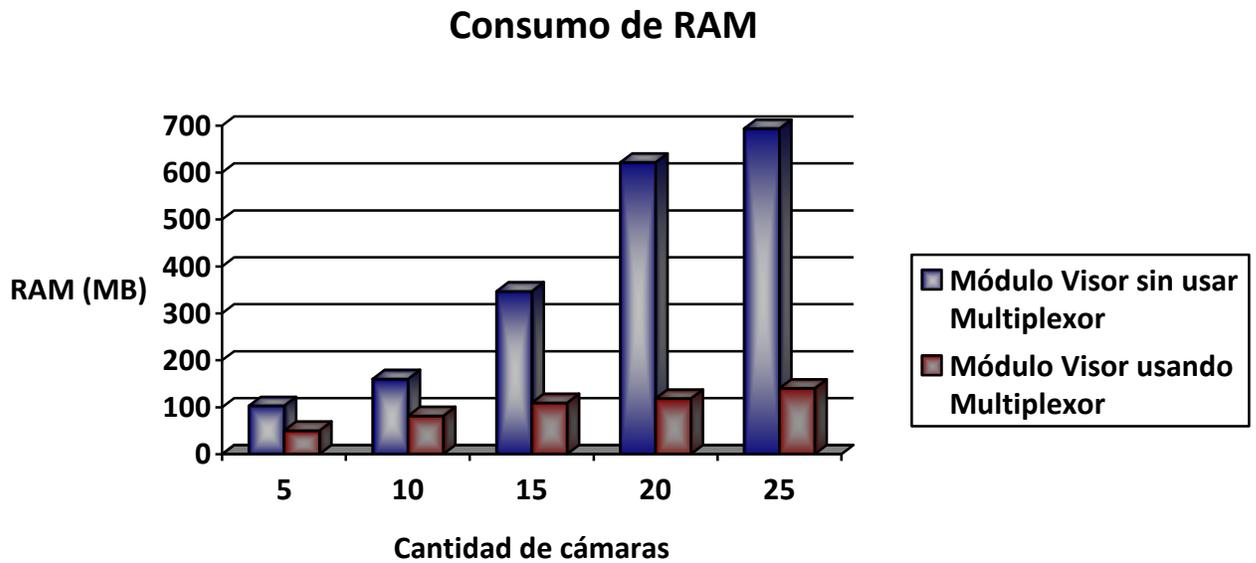


Fig. 9: Consumo de RAM (MB) en el sistema Xilema Suria, con y sin el uso del multiplexor.

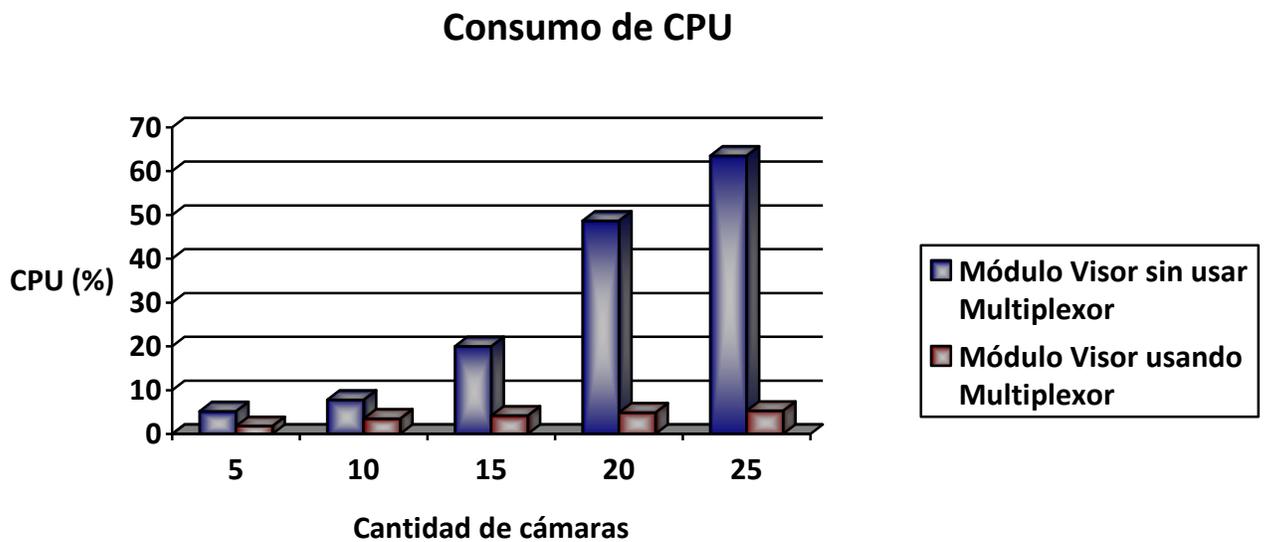


Fig. 10: Consumo de CPU (%) en el sistema Xilema Suria, con y sin el uso del multiplexor.

Capítulo 3: Implementación y Prueba

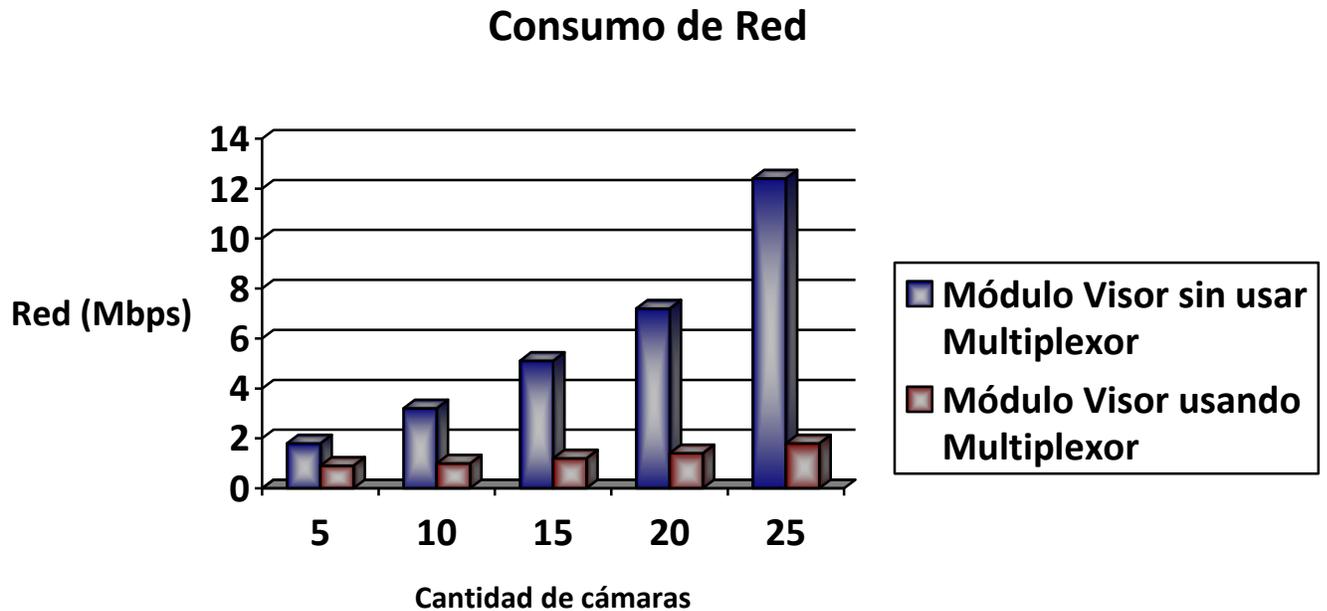


Fig. 11: Consumo de Red (Mbps) en el sistema Xilema Suria, con y sin el uso del multiplexor.

3.2.3 Prueba de Aceptación

Las pruebas de aceptación se realizan cuando se implementa un software a medida para un cliente, permitiendo que este valide todos los requisitos. La mayoría de los desarrolladores de productos de software llevan a cabo un proceso denominado prueba alfa y beta para detectar errores que sólo el usuario final puede descubrir. La alfa se lleva a cabo por un cliente, con el desarrollador como observador del usuario, registrando los errores y los problemas de uso en un entorno controlado. La beta se lleva a cabo por los usuarios finales en un entorno de trabajo real. El usuario registra todos los problemas (reales o imaginarios) e informa al desarrollador cada cierto tiempo (Pressman, 2010). Se decidió realizar una prueba de aceptación de tipo alfa al multiplexor de video IP en el laboratorio del proyecto Xilema Suria, en el centro de desarrollo GEySED. El Acta de Aceptación correspondiente a esta prueba puede encontrarse en el Anexo V del presente trabajo.

Capítulo 3: Implementación y Prueba

Conclusiones parciales

En este capítulo se definieron temas relacionados con los procesos de implementación y prueba del multiplexor de flujos de video provenientes de cámaras IP para el sistema Xilema Suria, arribándose a las siguientes conclusiones:

- ✓ Los diagramas de despliegue y de componentes de implementación, proporcionaron una vista de la implementación del sistema garantizando una descripción detallada de su estructura.
- ✓ Se probó que el software desarrollado cumple con la calidad requerida para ser integrado al sistema de video vigilancia Xilema Suria, proporcionándole menor consumo de red en la transmisión de los flujos de video y mejoras en el aprovechamiento del hardware de las estaciones de monitorización.

Conclusiones Generales

Conclusiones Generales

Con el presente trabajo de diploma se logró dar cumplimiento a las tareas de la investigación propuestas, logrando obtener como resultado un multiplexor de flujos de video provenientes de cámaras IP para el sistema Xilema Suria. La investigación permitió arribar a las siguientes conclusiones:

- ✓ La caracterización y revisión del estado del arte de los multiplexores de video, permitió afirmar que las soluciones que existen hoy, de alguna manera tributan a la investigación, pero no resuelven la problemática planteada, expresándose la necesidad del desarrollo de la propuesta.
- ✓ Las herramientas y tecnologías utilizadas en el desarrollo de la aplicación son las definidas por el proyecto Xilema Suria, favoreciendo con esto la integración entre el multiplexor y el sistema de video vigilancia, así como futuras actualizaciones del sistema.
- ✓ La utilización del patrón arquitectónico N-Capas permitió dividir el componente en dos capas: Comunicación y Negocio, permitiendo el desarrollo de la aplicación en varios niveles, lo que permite modificar la implementación de una capa sin afectar al resto del sistema.
- ✓ Las pruebas realizadas a la aplicación desarrollada demuestran que la herramienta cumple con las funcionalidades exigidas y está apta para ser integrada al sistema de video vigilancia Xilema Suria.
- ✓ Se logró desarrollar una aplicación capaz de capturar, multiplexar y transmitir los flujos de video provenientes de las cámaras IP, arrojando como resultado, la disminución del tráfico de datos en la red, así como el volumen de procesamiento en el hardware de las estaciones de monitorización.

Recomendaciones

Recomendaciones

Para mejorar el resultado y el impacto de la investigación se recomienda:

- ✓ Desarrollar en el módulo Visor una interfaz gráfica para interactuar con el multiplexor de video IP.
- ✓ Luego de finalizar la implementación del sistema Xilema Suria, se propone probar el uso del multiplexor de video IP con los módulos Analytic y Grabador, ya que los mismos tienen un comportamiento similar al módulo Visor en lo referido a la obtención de los flujos de video.
- ✓ Investigar una nueva tecnología que permita disminuir la latencia del sistema al transmitir los flujos de video multiplexados.

Referencias Bibliográficas

Referencias Bibliográficas

Álvarez, Miguel Angel. 2001. *¿Qué es Streaming?* San Juan : s.n., 2001.

Bass Electronics Incorporation. 2006. BassCam. Control Room Components and Accessories. [En línea] Bass Electronics, Inc, 2006. [Citado el: 19 de Noviembre de 2014.] <http://www.basscam.net>.

Bass, Len, Clements, Paul y Kazman, Rick. 2003. *Software Architecture in Practice, 2nd Edition.* New York : Addison Wesley, 2003.

Bates, Regis J y Bates , Marcus. 2007. *Voice and Data Communications.* New York : McGraw-Hill Computer - Communications Series, 2007. ISBN 9780072257328.

Bernal, Javier Hugo. 2010. *Análisis e implementación de un sistema de video.* Ciudad México : s.n., 2010.

Bond, Greg y Haldas, Mike. 2010. CCTV Camera Pros. CCTV Mux Grid Display Setup for VM-16RT. [En línea] Febrero de 2010. [Citado el: 27 de Noviembre de 2014.] <http://www.cctvcamerapros.com/>.

Bond, Greg y Haldas, Mike. 2009. CCTV Camera Pros. CCTV Video Multiplexers. [En línea] Octubre de 2009. [Citado el: 24 de Noviembre de 2014.] <http://www.cctvcamerapros.com>.

C++. 2014. cplusplus. [En línea] 2014. [Citado el: 21 de Enero de 2015.] <http://www.cplusplus.com/>.

Cabrera, Raúl y Salas, Nathalie María. 2013. *Subsistema distribuido de transmisión de contenido audiovisual.* La Habana : UCI, 2013.

Case, Albert F. 2001. *Computer-aided software engineering (CASE): technology for improving software development productivity.* New York : s.n., 2001. ISSN: 0095-0033.

Cisco Systems Incorporation. 2010. Cisco IPTV Head-end Solution - Cisco. *Cisco Systems, Inc. Web Site.* [En línea] Cisco Systems, Inc., Octubre de 2010. [Citado el: 20 de Noviembre de 2014.] <http://www.cisco.com>.

DATYS. 2011. DATYS Tecnología & Sistemas. [En línea] DATYS, 2011. [Citado el: 20 de Enero de 2015.] <http://www.datys.cu>.

Referencias Bibliográficas

Forouzan, Behrouz A. 2008. *Data Communications AND Networking Fifth Edition*. New York : McGraw-Hill, 2008. ISBN 978-0-07-337622-6.

Fowler, Martin, y otros. 2002. *Patterns of Enterprise Application Architecture* . Londres : Addison Wesley, 2002. ISBN:0-321-12742-0 .

Fritzler, Luciano. 2011. *Entorno de Desarrollo Integrado*. Buenos Aires : s.n., 2011.

Gamma, Erich, y otros. 2004. *Design Patterns*. Zúrich : s.n., 2004.

iMatix. 2014. ØMQ - The Guide. [En línea] iMatix, 2014. [Citado el: 15 de Enero de 2015.] <https://github.com/imatix/zguide>.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2005. *El Lenguaje Unificado de Modelado*. Madrid : Pearson Educacion, S.A, 2005. ISBN: 84-7829-042-2.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. *El Proceso Unificado de Desarrollo de Software*. Madrid : Pearson Educacion, S.A, 2000. ISBN: 84-7829-036-2.

Juristo, Natalia, Moreno, Ana M. y Vegas, Sira. 2006. *Técnicas de Evaluación de Software v12.0*. Madrid : s.n., 2006.

Kennelly, Arthur E. 2010. *Biographical Memoir of George Owen Squier*. s.l. : National Academy of Sciences, 2010.

Kruegle, Herman. 2011. *CCTV Surveillance: Video Practices and Technology*. Berlín : Butterworth-Heinemann, 2011. ISBN 9780080468181.

Lara García, Marisol. 2013. *Pruebas de Software*. México DF : s.n., 2013.

Larman, Craig. 2003. *Modelo del Dominio*. Burnaby : s.n., 2003.

Larman, Craig. 2000. *UML y Patrones*. Burnaby : s.n., 2000.

Leal, Myriam Yadira, Leal, Yuli Carolina y Medina, Leydi Carolina. 2011. *Arquitectura Cliente-Servidor*. Cundinamarca : Universidad de Cundinamarca, 2011.

Referencias Bibliográficas

Masinter, L., Berners-Lee, T. y McCahill, M. 2010. *Uniform Resource Locators (URL)*. San Francisco : s.n., 2010. RFC 1738.

Microsoft Corporation. 2011. Microsoft Corporation. [En línea] Microsoft Corporation, 2011. [Citado el: 14 de Enero de 2015.] <http://www.microsoft.com>.

Montagu, Arturo F. 2010. *Cultura Digital, Comunicación y Sociedad*. Buenos Aires : Paidós Ibérica, 2010.

Novoa, Pablo Montero. 2007. *Servidor modular de streaming*. Madrid : s.n., 2007.

Pantos, Roger y May, William. 2014. *HTTP Live Streaming*. Cupertino, California : Apple Inc., 2014.

Pressman, Roger S. 2010. *"Software Engineering"*. 7ma. New York : Higher Education, 2010.

Qt Project. 2013. Qt Creator 3.0 Manual. [En línea] Qt Project, 2013. [Citado el: 15 de Enero de 2015.] <http://qt-project.org/doc/qtcreator-3.0/>.

Qt. 2014. Qt. [En línea] 2014. [Citado el: 25 de Octubre de 2014.] <http://www.qt.io/>.

QtGStreamer. 2014. QtGStreamer. [En línea] 8 de Julio de 2014. <http://gstreamer.freedesktop.org>.

Quesada, Alma Evangelina. 2008. Frameworks-Webs. [En línea] 2008. [Citado el: 5 de Diciembre de 2014.] <http://www.freewebs.com>.

Reynoso, Carlos y Kicillof, Nicolás . 2004. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. Buenos Aires : Universidad de Buenos Aires, 2004.

Santos , Jorge Chávez. 2007. *Protocolos*. Lima : s.n., 2007.

Schulzrinne, Henning, Rao, Anup y Lanphier, Robert. 1998. *Real Time Streaming Protocol (RTSP)*. California : s.n., 1998. RFC 1889.

Sommerville, Ian. 2005. *Ingeniería de Software.Séptima Edición*. Madrid : Pearson Education, 2005. ISBN: 84-7829-074-5.

Sun, Ming-Ting y Reibman, Amy R. . 2000. *El video comprimido a través de redes*. Washington : s.n., 2000. ISBN:978-1-4398-7331-1.

Referencias Bibliográficas

Taymans, Wim , y otros. 2012. *GStreamer Application Development*. Barcelona : s.n., 2012.

Torres , Daniel Oscar. 2009. *Evolución y tendencia de la tecnología streaming en Internet*. Madrid : s.n., 2009.

Valeriano, A. J. 2010. *¿Qué es y cómo funciona una Cámara IP?* Oley Valley, Pennsylvania : s.n., 2010.

Bibliografía

Bibliografía

Álvarez, Miguel Angel. 2001. *¿Qué es Streaming?* San Juan : s.n., 2001.

Forouzan, Behrouz A. 2008. *Data Communications AND Networking Fifth Edition.* New York : McGraw-Hill, 2008. ISBN 978-0-07-337622-6.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2005. *El Lenguaje Unificado de Modelado.* Madrid : Pearson Educacion, S.A, 2005. ISBN: 84-7829-042-2.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. *El Proceso Unificado de Desarrollo de Software.* Madrid : Pearson Educacion, S.A, 2000. ISBN: 84-7829-036-2.

Larman, Craig. 2003. *Modelo del Dominio.* Burnaby : s.n., 2003.

Larman, Craig. 2000. *UML y Patrones.* Burnaby : s.n., 2000.

Novoa, Pablo Montero. 2007. *Servidor modular de streaming.* Madrid : s.n., 2007.

Pressman, Roger S. 2010. *"Software Engineering".* 7ma. New York : Higher Education, 2010.

Qt Centre. 2014. Qt Centre. [En línea] Qt Centre, 2014. <http://www.qtcentre.org>.

Qt Project. 2013. Qt Creator 3.0 Manual. [En línea] Qt Project, 2013. <http://qt-project.org/doc/qtcreator-3.0/>.

QtGStreamer. 2014. QtGStreamer. [En línea] 8 de Julio de 2014. <http://gststreamer.freedesktop.org>.

Reynoso, Carlos y Kicillof, Nicolás . 2004. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft.* Buenos Aires : Universidad de Buenos Aires, 2004.

Sommerville, Ian. 2005. *Ingeniería de Software.Séptima Edición.* Madrid : Pearson Education, 2005. ISBN: 84-7829-074-5.

Taymans, Wim , y otros. 2012. *GStreamer Application Development.* Barcelona : s.n., 2012.

Anexos

Anexos

I. Entrevista realizada

Fecha: noviembre 2014

Entrevistados: Reynier Pupo Gómez, Rafael Leodán Cardero Álvarez y Rayner Pupo Gómez.

Preguntas:

1. ¿Considera usted importante el desarrollo de una aplicación que multiplexe los flujos de video? ¿Qué beneficios traería para el proyecto Xilema Suria una aplicación de este tipo?
2. ¿Qué requisitos considera que sean importantes a desplegar por la aplicación?

Resultados generales de la entrevista por preguntas:

1. Una aplicación como el multiplexor provee al sistema de video vigilancia Xilema Suria de un entorno más seguro y estable para la transmisión de los flujos de video emitidos por las cámaras IP. Además disminuye considerablemente el tráfico de red, evitando que se formen “cuellos de botella” en las salidas de los dispositivos de interconexión. A grandes rasgos algunos de los beneficios que traería para el proyecto Xilema Suria un multiplexor de video IP son:
 - ✓ Disminución del consumo de red, así como del impacto del sistema sobre los recursos de RAM y CPU en las estaciones de monitorización.
 - ✓ Facilidades en el despliegue del sistema de video vigilancia ya que este puede ser desplegado en estaciones de trabajo con menos prestaciones de las que se requieren actualmente.
 - ✓ Menor tiempo de respuesta en la transmisión de video ante múltiples conexiones a la misma cámara.
2. Algunos de los requisitos coincidentes mencionados por los entrevistados son:
 - ✓ Capturar flujos de video provenientes de las cámaras IP.
 - ✓ Multiplexar flujos de video.
 - ✓ Transmitir flujo de video multiplexado.
 - ✓ Detener multiplexión de video.

II. Descripción del Caso de Uso Detener Multiplexor

Tabla 6: Descripción del CU Detener multiplexor.

CU	Detener multiplexor.	
Objetivo	El objetivo de este caso de uso es detener el proceso de multiplexión para un Visor determinado.	
Actores	Visor.	
Resumen	El caso de uso inicia cuando se selecciona la opción Cerrar en el Visor donde se está mostrando el flujo de video multiplexado, terminando así el CU.	
Complejidad	Media.	
Prioridad	Auxiliar.	
Precondiciones	Proceso de multiplexión iniciado. Visor seleccionado.	
Postcondiciones	Multiplexión finalizada.	
Flujo de eventos		
Flujo básico: Detener multiplexor.		
	Actor	Sistema
1.	El Visor del sistema Xilema Suria envía al multiplexor de video IP el identificador del Visor que ha sido detenido.	
2.		El multiplexor recibe el identificador y finaliza el proceso de multiplexión para el Visor al que pertenece. Termina el CU.
Flujos alternos.		
1a. Identificador equivocado.		
	Actor	Sistema
1a		El multiplexor recibe el identificador y lo compara con los identificadores de las transmisiones que está realizando. Si el identificador recibido no coincide con el de ninguna transmisión, el multiplexor no realiza ninguna acción. Termina el CU.
Relaciones	CU Incluidos	No procede.
	CU Extendidos	No procede.

Referencias	RF 4, RnF 6, RnF 7, RnF 9, RnF 10.
-------------	------------------------------------

III. Diagrama de secuencia del Caso de Uso Detener Multiplexor.

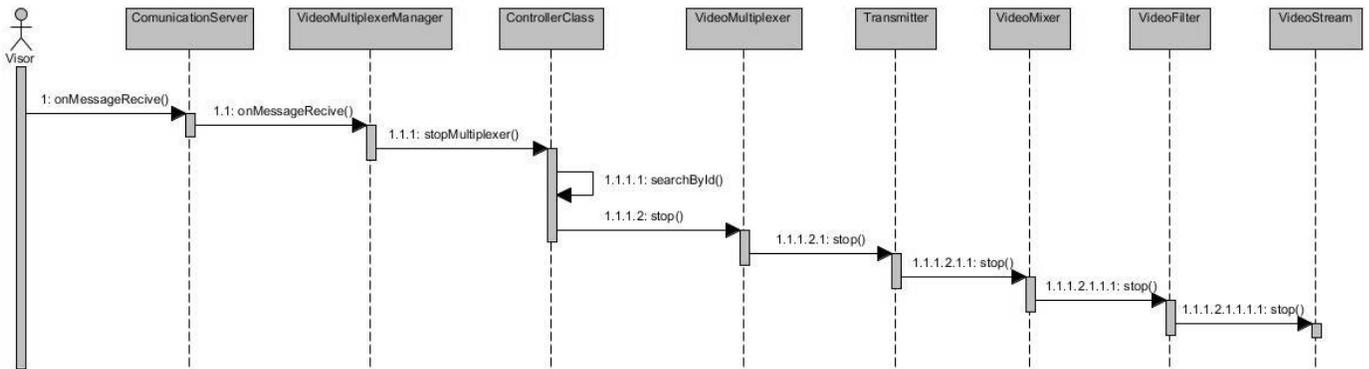


Fig. 12: Diagrama de Secuencia del CU Detener Multiplexor.

IV. Diagrama de Clases del Diseño

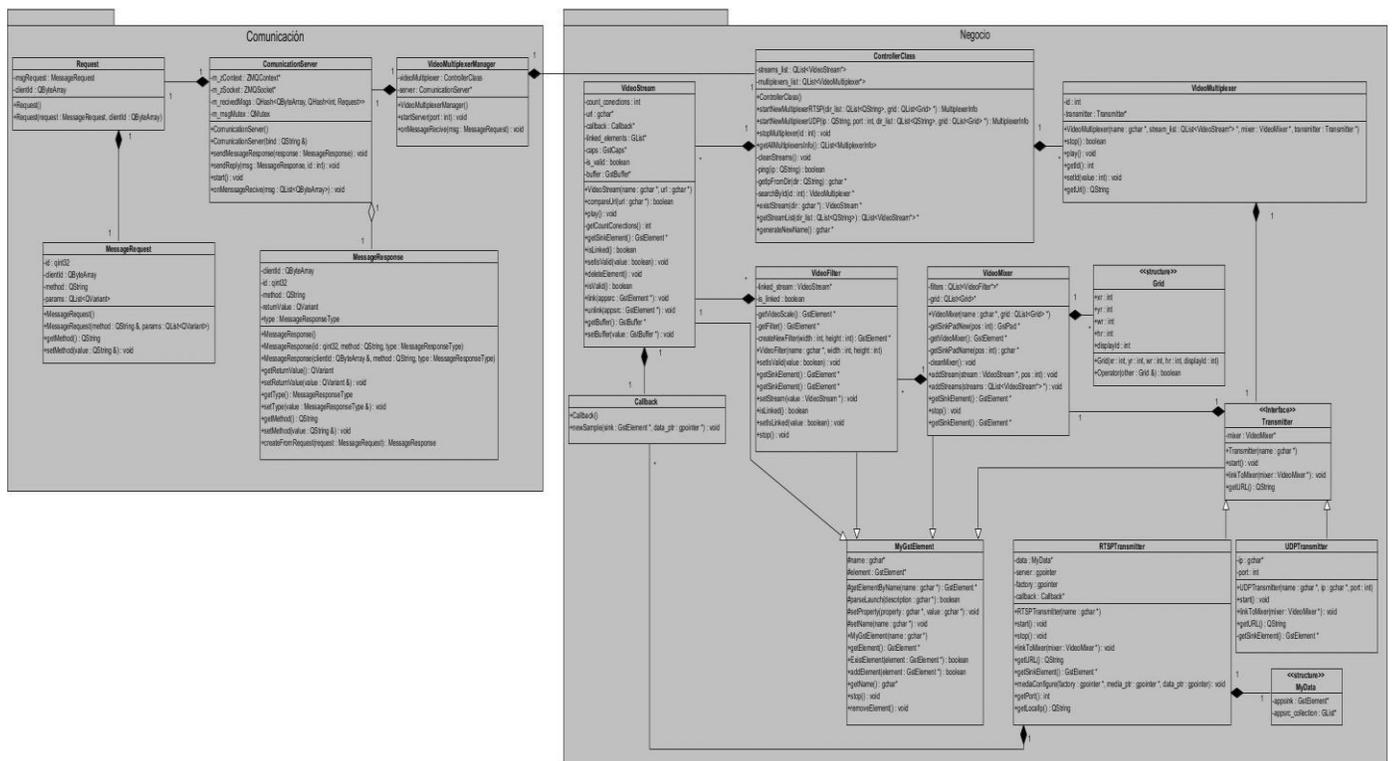


Fig. 13: Diagrama de Clases del Diseño.

V. Acta de Aceptación

UCI Universidad de las Ciencias Informáticas

Acta de aceptación

ACTA DE ACEPTACIÓN

En cumplimiento con la fase de desarrollo y en función de la ejecución del proyecto: Video Vigilancia Xilema Suria para el Departamento de Desarrollo de Componentes del Centro de Geoinformática y Señales Digitales (GEySED), se hace entrega de los productos que se relacionan a continuación:

- Multiplexor de flujos de video provenientes de cámaras IP (código fuente).
- Artefactos ingenieriles elaborados en el proceso de desarrollo.

La Parte Cliente, luego de haber revisado los productos de trabajo determina que: se acepta la aplicación desarrollada, la cual aporta una funcionalidad muy importante para el proyecto. Utilizando el multiplexor de videos desarrollado, se podrá disminuir considerablemente el consumo de recursos de hardware actuales del sistema Xilema Suria en cuanto a la transmisión, captura y representación de los flujos de videos transmitidos.

Entrega	Recibe
Nombre y apellidos: Reidel Morales Toledo Serguey Caballero Sarduy	Nombre y apellidos: Ing. Reynier Pupo Gómez
Cargo: Tesisistas	Cargo: Jefe de Proyecto (Xilema Suria)
Firma: 	

UCI Universidad de las Ciencias Informáticas
Centro de Desarrollo de Geoinformática y Señales Digitales
Fecha: 19/06/2015

Fig. 14: Acta de Aceptación del Multiplexor de video IP por parte del proyecto de video vigilancia Xilema Suria.

Glosario de Términos

Glosario de Términos

Application Programming Interface (API): Interfaz de Programación de Aplicaciones.

Computer Aided Software Engineering (CASE): Ingeniería de Software Asistida por Computación.

Closed Circuit Television (CCTV): Circuito cerrado de televisión. Tecnología de video vigilancia visual diseñada para supervisar una diversidad de ambientes y actividades.

Cuellos de botella (bottleneck): Es cuando se realizan muchas solicitudes pero no pueden ser atendidas al mismo tiempo, quedando en una fila de espera hasta llegar un punto que quién está atendiendo las solicitudes no puede atender más ninguna, saturándose y terminando el proceso.

Framework (Marco de Trabajo): Define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

Hypertext Transfer Protocol (HTTP): Protocolo de Transferencia de Hipertexto.

Internet Protocol (IP): Es un protocolo de comunicación de datos digitales clasificado funcionalmente en la Capa de Red según el modelo internacional OSI.

Local Area Network (LAN): Red de Área Local.

Open System Interconnection (OSI): Modelo de interconexión de sistemas abiertos (ISO/IEC 7498-1).

Real Time Messaging Protocol (RTMP): Protocolo de Mensajería en Tiempo Real. Proporciona un servicio de mensajes a través de un flujo de transporte fiable, destinado a transportar corrientes paralelas de video, audio y mensajes de datos, con información de temporización asociada.

Real Time Streaming Protocol (RTSP): Protocolo de Flujo en Tiempo Real.

Transmission Control Protocol (TCP): Protocolo de Control de Transmisión.

Unicast: Conexión donde el cliente establece un enlace con el servidor y recibe o envía una cadena de datos independiente a la comunicación que se realiza con otros clientes.