

# Universidad de las Ciencias Informáticas

Facultad 6



## “Módulo Recuperador para Xilema-Suria 2.0”

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

### **Autores:**

Arianna López Alfonso

Rafael Bermudez Martell

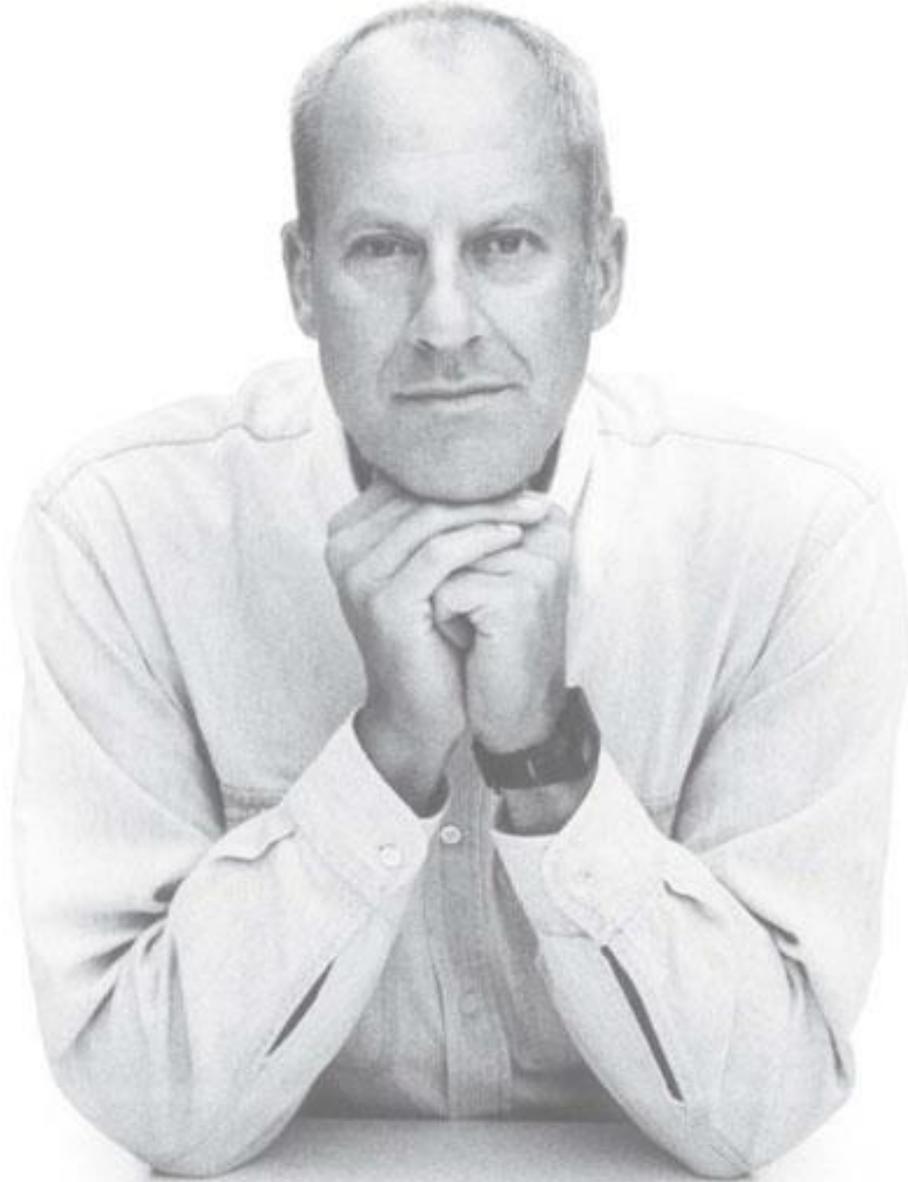
### **Tutor:**

Ing. Yenisel Tirado Gutiérrez

Ing. Guillermo Luzua Farias

La Habana, junio del 2016

“Año 58 de la Revolución”



*“Si tienes una pasión, si realmente crees en algo, esfuérzate y podrás hacer posibles las cosas con las que has soñado.”*

*Norman Foster*



**DECLARACIÓN DE AUTORÍA**

Declaramos ser los únicos autores del trabajo **Módulo Recuperador para Xilema-Suria 2.0** y autorizamos a la Universidad de las Ciencias Informáticas a hacer el uso que estime pertinente con el mismo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

**Firma del autor**

Arianna López Alfonso

---

**Firma del autor**

Rafael Bermudez Martell

---

**Firma del Tutor**

Ing. Yenisel Tirado Gutiérrez

---

**Firma del Tutor**

Ing. Guillermo Luzua Farías

**Nombre y Apellidos:** Ing. Guillermo Luzua Farías.

**Correo electrónico:** gluzua@uci.cu

**Año de graduado:** 2012.

**Nombre y Apellidos:** Ing. Yenisel Tirado Gutiérrez.

**Correo electrónico:** ytgutierrez@uci.cu

**Año de graduado:** 2015.

*Arianna:*

*A mi madre Aiza por su apoyo y confianza,  
a mi padre Evens por ser el impulsor de este sueño y  
a Arian, mi pedacito de vida, que a pesar de llegar a mitad de  
mi carrera fue el motorcito de arranque para llegar hasta aquí.*

*Rafael:*

*A mis padres por todo su amor y apoyo incondicional.  
A mi familia por siempre creer en mí.*

*Arianna:*

A la vida por estos años de carrera universitaria, donde he conocido a personas increíbles, desde mis profesores hasta mis compañeros de aula y cuarto.

A mis padres, porque sin ellos no creo que hubiese podido llegar hasta aquí.

A mi familia, incluyendo a mi familia santiaguera porque siempre han creído en mí y por su apoyo incondicional.

A mis bellezas: Arian, Diegú, Rony y Kemily, por darle el toque mágico a mi vida, porque siempre ponen una sonrisa en mi cara pese al momento.

A mi compañero de tesis Bermudez, por ser constante en el trabajo conmigo y por ayudarme.

A mi "mami UCI", por ser compañera, amiga, hermana, confidente y tutora, por compartir tantos momentos juntas y por prepararme para este momento tan importante.

A mi otro tutor, Guillermo, por su ayuda, por todo su apoyo y sus enseñanzas.

A todo el tribunal que con sus críticas y sugerencias nos ayudaron a realizar esta tesis, en especial a Rafael Leodan.

A los compañeros de aula, por su amistad, porque juntos compartimos momentos increíbles que nunca olvidaré.

Agradecer también a todas las amistades que he hecho durante la carrera, porque hicieron de la UCI una experiencia inolvidable.

A la UCI por la oportunidad de superarme como persona.

A todas las personas que me han apoyado a lo largo de la carrera y que han hecho posible esta tesis, a todos en general muchísimas gracias.

### *Rafael:*

A mis padres que siempre me han apoyado y aconsejado a cada paso del camino, que me han enseñado a levantarme y ser fuerte.

A mis hermanos y mi familia por siempre creer en mí aun cuando yo no lo hago.

A Denisse por ser un pilar en mi vida.

A todos mis amigos, los que están y los que no han podido estar.

A mi compañera de tesis Arianna por toda su preocupación.

A nuestros tutores Yení y Guillermo por su ayuda y preocupación, por estar siempre dispuestos y por todo su apoyo y enseñanzas.

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 siempre dispuestos a contestar preguntas o aclarar dudas.

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

A todos aquellos que han formado parte de mi estancia en esta universidad y a quienes agradezco cada minuto compartido.

Gracias.

## Resumen

La Universidad de las Ciencias Informáticas cuenta con un centro dedicado al desarrollo de soluciones de video vigilancia, su producto insignia es el software Xilema-Suria 2.0. Actualmente está compuesto por seis tipos de módulos, los cuales brindan funcionalidades que satisfacen las necesidades de las empresas e instituciones cubanas. El sistema no posee un módulo que permita la recuperación de los videos almacenados, siendo el objetivo de la presente investigación el desarrollo de un Módulo Recuperador para Xilema-Suria 2.0. Con el cual se pretende disminuir el tiempo empleado en las actividades de búsqueda y visualización de las grabaciones, agilizando y facilitando la interacción con el Sistema de Video Vigilancia. Para su desarrollo fue utilizada la metodología AUP-UCI, utilizando la herramienta CASE *Visual Paradigm* para el modelado y el lenguaje de programación C++ para la implementación. Para evaluar los resultados de la investigación se realizaron las pruebas de caja negra e integración. Demostrando que el módulo cumple con los requisitos especificados y que cumple con la calidad requerida para ser integrado al Sistema de Video Vigilancia Xilema-Suria 2.0.

**Palabras Claves:** grabaciones, módulo recuperador, recuperación, Sistema de Video Vigilancia, visualización.

## **Abstract**

The University of Information Science has a center dedicated to the development of video surveillance solutions, Xilema-Suria 2.0 Software is its flagship product. The software comprises six types of modules, which provide functions that meet the needs of Cuban companies and institutions. The current system does not have a module that allows the retrieval of stored videos. Therefore, this research aims to develop a recovery module for Xilema-Suria 2.0 and intends to reduce the time required for both research as well as the time taken to view a recording. Thus, expediting and facilitating the interaction with the video surveillance system. The AUP-UCI methodology was used for its development while employing the CASE tool Visual Paradigm for modeling, programming language C++ was used for implementation. To evaluate the research, results were performed through black box and integration testing, proving that the module meets the specified conditions and quality required to be integrated with the video surveillance system Xilema-Suria 2.0.

**Keywords:** recording, recovery module, retrieval, video surveillance system, view.

## Índice General

<b>Introducción</b> .....	1
<b>Capítulo 1. “Fundamentación teórica sobre los procesos de recuperación en sistemas de video vigilancia”</b> .....	6
1.1    Conceptos asociados a la investigación.....	6
1.1.1    Sistemas de video vigilancia .....	6
1.1.2    Recuperación de Información .....	6
1.1.3    Recuperación de videos grabados.....	7
1.2    Evolución de los sistemas de video vigilancia .....	7
1.3    Evolución del proceso de recuperación de grabaciones .....	10
1.4    Descripción de la situación problemática .....	12
1.5    Revisión de sistemas nacionales e internacionales .....	13
1.5.1    Sistema de video vigilancia <i>AXIS Camera Station</i> .....	13
1.5.2    Sistema de video vigilancia <i>SavVi</i> .....	14
1.5.3    Sistema de video vigilancia <i>Xyma Safe Vision</i> .....	14
1.5.4    Sistema de video vigilancia de la Dirección de Comunicación Institucional UCI .....	14
1.5.5    Análisis de las soluciones similares. ....	14
1.6    Metodología, herramientas y tecnologías a utilizar .....	15
1.6.1    Metodología de desarrollo de software: AUP-UCI.....	15
1.6.2    Lenguaje de modelado: UML.....	16
1.6.3    Herramienta CASE: <i>Visual Paradigm for UML</i> .....	16
1.6.4    Lenguaje de Programación: C++ .....	17
1.6.5 <i>Framework</i> de desarrollo: Qt.....	17
1.6.6    Entorno de Desarrollo Integrado: Qt Creator.....	17
1.6.7    Biblioteca: ZMQ .....	18
1.7    Conclusiones del capítulo .....	18
<b>Capítulo 2. “Características del Módulo Recuperador para Xilema-Suria 2.0”</b> .....	20
2.1    Modelo de dominio .....	20
2.1.1    Diagrama del modelo de dominio .....	20
2.1.2    Descripción de los conceptos que intervienen en el dominio del problema .....	21
2.1.3    Descripción del flujo del modelo de dominio .....	21
2.2    Especificación de los requisitos .....	21

2.2.1	Requisitos funcionales .....	21
2.2.2	Requisitos no funcionales .....	23
2.3	Modelo de casos de uso del sistema .....	24
2.3.1	Casos de uso del sistema .....	24
2.3.2	Descripción del actor del sistema .....	26
2.3.3	Diagrama de caso de uso del sistema .....	26
2.3.4	Descripción de los casos de uso del sistema .....	27
2.4	Descripción de la propuesta de solución .....	36
2.5	Arquitectura del software .....	37
2.5.1	Arquitectura en capas .....	37
2.6	Diagrama de Clases del Diseño .....	38
2.7	Patrones de Diseño .....	40
2.7.1	Patrones GRASP empleados .....	40
2.7.2	Patrones GoF empleados .....	41
2.8	Conclusiones del capítulo .....	41
<b>Capítulo 3: “Implementación y prueba del Módulo Recuperador para Xilema-Suria 2.0” .....</b>		<b>43</b>
3.1	Modelo de Implementación .....	43
3.1.1	Diagrama de componentes de implementación .....	43
3.1.2	Diagrama de despliegue .....	45
3.2	Código fuente .....	46
3.2.1	Estándares de codificación .....	46
3.3	Pruebas .....	47
3.3.1	Prueba de Caja Negra: Partición equivalente .....	47
3.3.2	Prueba de Integración .....	55
3.4	Conclusiones del capítulo .....	57
<b>Conclusiones Generales .....</b>		<b>58</b>
<b>Recomendaciones .....</b>		<b>59</b>
<b>Bibliografía .....</b>		<b>60</b>

## Índice de Tablas

Tabla 1. Evolución de los sistemas de video vigilancia .....	11
Tabla 2. Características de las soluciones analizadas .....	15
Tabla 3. Actor del sistema.....	26
Tabla 4. Descripción del CU: Realizar Búsquedas .....	28
Tabla 5. Descripción del CU: Administrar conexión con el Gestor.....	32
Tabla 6. Descripción del CU: Realizar Operaciones con el Reproductor.....	33
Tabla 7. Descripción del caso de prueba: CU Realizar búsqueda (Sección 1) .....	49
Tabla 8. Descripción del caso de prueba: CU Realizar búsqueda (Sección 2) .....	50
Tabla 9. Descripción del caso de prueba: CU Realizar operaciones con el reproductor.....	51
Tabla 10. Descripción del caso de prueba: Administrar Conexión con el Gestor.....	53

## Índice de Figuras

Figura 1. Sistema de video vigilancia completamente analógico. (Axis, 2014) .....	8
Figura 2. Sistema de video vigilancia analógico usando DVR. (Axis, 2014) .....	9
Figura 3. Sistema de video vigilancia analógico usando DVR de red. (Axis, 2014) .....	9
Figura 4. Sistema de video vigilancia usando servidor de video Axis. ....	10
Figura 5. Sistema de video vigilancia usando cámaras de red Axis. (Axis, 2014).....	10
Figura 6. Modelo del dominio del Módulo Recuperador para Xilema-Suria 2.0 .....	20
Figura 7. Diagrama de casos de uso del Módulo Recuperador para Xilema-Suria 2.0.....	27
Figura 8. Arquitectura del Módulo Recuperador para Xilema-Suria 2.0.....	38
Figura 9. Diagrama de Clases del Módulo Recuperador para Xilema-Suria 2.0.....	39
Figura 10. Diagrama de componentes del Módulo Recuperador para Xilema-Suria 2.0 .....	44
Figura 11. Diagrama de Despliegue del Módulo Recuperador para Xilema-Suria 2.0 .....	46
Figura 12. Gráfico de NC de las pruebas de caja negra.....	55

## Introducción

La vigilancia es el monitoreo del comportamiento, actividades, u otra información cambiante, por lo general de personas con el propósito de influir, gestionar, dirigir, o protegerlos. La video vigilancia según la Real Academia de la Lengua Española (RAE), se define como “vigilancia a través de un sistema de cámaras, fijas o móviles” (RAE, 2014). Los sistemas de vigilancia por video han tenido una evolución acelerada debido a tres motivos principales; el desarrollo tecnológico, la demanda de mayor nivel de seguridad y el estudio de técnicas de análisis de video (García, 2010). El uso más común y para lo que fueron concebidos en un principio es para la detección de intrusiones en espacios, cerrados o abiertos. Sin embargo, las aplicaciones de la video vigilancia IP se han ido incrementando a medida que el software ha avanzado en prestaciones. En este sentido, ya es posible realizar control de accesos, tanto de personas como de vehículos; o el control de procesos en cadenas de montaje, reforzando la prevención de riesgos laborales.

Los sistemas de video vigilancia comenzaron siendo totalmente analógicos y han evolucionado paulatinamente digitalizándose. En sus inicios las grabaciones se realizaban a través de los videograbadores, que contaban con el *cassette* como medio de almacenamiento. Las grabaciones solo duraban un par de horas, lo cual traía como consecuencia la necesidad de gran cantidad de *cassettes* para dar respaldo al sistema. Además, la búsqueda de una escena específica en los mismos era muy compleja porque se debía ir reproduciendo archivos unitariamente y muchas veces es tedioso localizar la parte que realmente se busca.

En la actualidad, estos sistemas utilizan cámaras y servidores digitales para la grabación de videos y su almacenamiento. Conocida como video vigilancia IP (del inglés *Internet Protocol*, protocolo de internet), es una tecnología de vigilancia visual que combina los beneficios analógicos de los Sistemas de Circuitos Cerrados de Televisión (CCTV) tradicionales con las ventajas digitales de las redes de comunicación IP, permitiendo la supervisión local y/o remota de imágenes y audio, así como el tratamiento digital de las imágenes, para aplicaciones como el reconocimiento de matrículas o reconocimiento facial entre otras. El despliegue resulta más sencillo y económico que un CCTV, puesto que aprovecha la red informática, es decir, el mismo cableado que se emplea para la comunicación de datos, acceso a Internet o correo electrónico, sin necesidad de desplegar una infraestructura de cableado coaxial específica para nuestra red de video vigilancia. Las búsquedas de algún evento o video específico se simplifican, ya que permite buscar a través de la línea de tiempo o incluso por criterios de búsquedas.

A nivel internacional, existen varias compañías que han lanzado al mercado mundial soluciones de video vigilancia sumamente efectivas como es el caso de Axis con el desarrollo del *AXIS Camera Station* o de Panasonic que ofrece un Sistema de Video Vigilancia específico para el kit de cámaras IP que comercializa. A nivel nacional se encuentra la empresa de desarrollo de tecnologías y sistemas DATYS, la cual cuenta con un sistema de video protección profesional basado en tecnología IP llamado *Xyma Safe Vision*.

En la Universidad de las Ciencias Informáticas (UCI), específicamente en el centro de Geoinformática y Señales Digitales (GEYSED) se desarrolló el Sistema de Video Vigilancia Xilema-Suria 2.0. El cual posee una arquitectura que le permite adaptarse a los modelos de cámaras para las cuales se desarrollan los *plugin*<sup>1</sup> de integración. Además, cuenta con un video sensor integrado que detecta movimiento por cámaras y emite alertas sonoras y visuales en función de la alarma activada.

El sistema está compuesto por seis módulos que interactúan entre sí para garantizar la monitorización de cualquier entorno de manera eficiente, utilizando cámaras IP de distintos fabricantes. Además, está diseñado con la particularidad de que cada módulo funciona de manera autónoma, siguiendo una arquitectura en pizarra. El **Módulo de Administración** es el encargado de gestionar toda la información del sistema; el **Módulo de Grabación** almacena los flujos capturados por las cámaras; el **Módulo de Autonomía** efectúa la gestión de los dispositivos de almacenamiento del sistema; el **Módulo Visor** permite visualizar los flujos de videos capturados por las cámaras; el **Módulo de Análisis** realiza el procesamiento a los flujos de videos que le sean asignados y el **Módulo Gestor**, encargado de orquestar todo el flujo de información en el sistema de manera segura.

Actualmente, el proceso de recuperación de videos grabados en Xilema-Suria 2.0 se realiza accediendo a los directorios de almacenamiento físico, donde se encuentran las grabaciones realizadas de forma diaria. Este proceso resulta engorroso, debido a que el acceso a las grabaciones de video se realiza mediante los exploradores nativos del sistema operativo (SO) donde se encuentre instalado el módulo de autonomía. Lo que propicia el acceso indebido a materiales restringidos, por lo tanto, compromete la integridad de los datos. Además, las grabaciones de las cámaras generan grandes volúmenes de información en cuanto a archivos de videos respecta. La recuperación de estos archivos en los dispositivos de almacenamiento tiende a demorarse y no siempre se obtiene la información de manera oportuna para los análisis pertinentes.

---

<sup>1</sup> *Plugin*: Complemento que añade una funcionalidad adicional o una nueva característica al software. (definicion, 2015)

A partir del análisis de la situación anteriormente descrita se identificó el siguiente **problema de investigación**: ¿Cómo facilitar el proceso de recuperación de las grabaciones realizadas en Xilema-Suria?

Se define como **objetivo general**, desarrollar un módulo para la recuperación de las grabaciones realizadas en el sistema Xilema-Suria 2.0.

Para brindar una solución acorde con la problemática planteada se centra el estudio de la investigación en el siguiente **objeto de estudio**: el proceso de recuperación de materiales audiovisuales. Acotando el **campo de acción** a las técnicas de recuperación de grabaciones en los sistemas de video vigilancia.

Para cumplir el objetivo general de la investigación se trazaron las siguientes **preguntas científicas**:

- ✓ ¿Cuáles son los elementos teóricos de los sistemas de video vigilancia?
- ✓ ¿Qué herramientas y tecnologías son apropiadas para el desarrollo del Módulo Recuperador para Xilema-Suria 2.0?
- ✓ ¿Cómo estructurar el proceso de desarrollo del Módulo Recuperador para Xilema-Suria 2.0?
- ✓ ¿Qué resultados se obtendrán al aplicar técnicas y/o métodos ingenieriles al módulo?

Para dar cumplimiento a las preguntas anteriores se hace necesario desarrollar las siguientes **tareas de investigación**:

1. Definición de los principales conceptos, características y tendencias actuales relacionadas con los sistemas de video vigilancia y en el proceso de recuperación de los mismos.
2. Caracterización de las soluciones existentes nacionales e internacionales.
3. Descripción de la metodología, las herramientas y tecnologías a utilizar durante el desarrollo del Módulo Recuperador para Xilema-Suria 2.0.
4. Especificación de los requisitos funcionales y no funcionales para definir las características del módulo a desarrollar.
5. Elaboración del diseño de la propuesta de solución para guiar el proceso de implementación.
6. Implementación del Módulo Recuperador para Xilema-Suria 2.0.
7. Ejecución de las pruebas de software para validar el correcto funcionamiento del módulo.

Para la realización de las tareas trazadas anteriormente, se utilizaron los siguientes **métodos de investigación**: (Hernández León, y otros, 2002)

## Métodos Teóricos

**Analítico - Sintético:** Permite la descomposición de un todo complejo en sus partes y cualidades. La síntesis, por su parte, establece la unión entre las partes, previamente analizadas y posibilita descubrir relaciones y características generales entre los elementos de la realidad. Se utiliza para realizar un estudio de la bibliografía referente a los diferentes conceptos asociados a la investigación, permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio.

**Histórico - Lógico:** El método histórico estudia la trayectoria real de los fenómenos y acontecimientos en el transcurso de su historia. El método lógico investiga las leyes generales del funcionamiento y desarrollo de los fenómenos. Este método se emplea para analizar la evolución de los conceptos asociados a la recuperación de videos, permitiendo formar una definición propia. Además, se utiliza para profundizar en el estudio de la evolución y desarrollo de los sistemas de video vigilancia y los procesos de recuperación de videos utilizados en los mismos, lo que permite ver en qué etapa se encuentran dichos sistemas y cómo realizan la recuperación de videos grabados.

**Modelación:** Se utiliza para la representación del modelado mediante diagramas. Esto es evidenciado en los artefactos que son generados durante el desarrollo del software, dando un mejor entendimiento de cómo implementar la solución.

### **Métodos Empíricos**

**Entrevista:** Se utiliza con el objetivo de entrevistar a algunos profesionales del proyecto para un mayor dominio del tema a investigar y para obtener algunas características y funcionalidades a tener en cuenta para el desarrollo del Módulo Recuperador para Xilema-Suria 2.0. Se efectuó el tipo de entrevista no estructurada ya que es más abierta que la estructurada, donde se establece el tema, pero no se lleva a cabo un cuestionario rígido.

**Observación:** Se emplea para realizar valoraciones y obtener información a partir de lo observado. Esto se manifiesta principalmente cuando se realizan observaciones sobre el funcionamiento de sistemas similares al módulo a desarrollar, dando una visión de cómo tiene que ser el sistema a realizar en su forma externa.

**Análisis documental:** Se utiliza en la revisión de la literatura especializada para consultar la información necesaria en el proceso de investigación, también para representar un documento y su contenido bajo una forma diferente de su forma original, con la finalidad de posibilitar su recuperación posterior e identificarlo.

Como **aporte práctico** de la investigación se desarrolló un módulo de recuperación de videos grabados para el Sistema de Video Vigilancia Xilema-Suria 2.0. El mismo apoya y facilita en gran medida la gestión de búsquedas de los videos que se graban en el sistema. Brinda varios criterios de búsqueda de los videos

en todas las zonas y cámaras que sean visibles para la persona que trabaje con el módulo. Además, permite tener integrada la herramienta de análisis de video, lo que facilita la identificación de eventos de interés en los videos, al igual que el control de la reproducción de los videos, temas de mucha utilidad para el análisis de hechos extraordinarios que queden registrado por las cámaras del sistema.

## **Estructura del trabajo de diploma**

El presente trabajo de diploma consta de **tres capítulos** donde se abarca todo el proceso investigativo realizado. En el **primer capítulo** se fundamentan términos técnicos y se analizan algunos sistemas de video vigilancia, sirviendo de base para la investigación. Además, se describen la metodología, herramientas y tecnologías a utilizar para el desarrollo del Módulo Recuperador para Xilema-Suria 2.0. El **segundo capítulo** abarca todo lo relacionado con el análisis y diseño del módulo a desarrollar, mostrando algunos diagramas para un mejor entendimiento. Por último, en el **tercer capítulo** se aborda el desarrollo de la solución, además se muestran el diagrama de despliegue y el de componentes de implementación. Conjuntamente se realizan las descripciones de las pruebas y se exponen los resultados de las mismas.

## **Capítulo 1. “Fundamentación teórica sobre los procesos de recuperación en sistemas de video vigilancia”**

En el presente capítulo se expondrán las principales definiciones relacionadas con la investigación. Además, se realizará un estudio del arte de los sistemas de video vigilancia existentes en Cuba y el mundo que realizan la recuperación de videos grabados. Por último, se describen las herramientas, metodología y tecnologías a utilizar en la realización del Módulo Recuperador para Xilema-Suria 2.0.

### **1.1 Conceptos asociados a la investigación**

Los conceptos abarcados en este epígrafe fueron seleccionados por su gran importancia en el estudio del campo de la video vigilancia. Además, es necesario conocer su significado para llegar a entender el tema tratado en la investigación.

#### **1.1.1 Sistemas de video vigilancia**

Los sistemas de video vigilancia supervisan, controlan y registran la actividad física dentro de un espacio determinado. Las imágenes e información generada en el sistema, es solo accesible por parte de una serie de usuarios con acceso al sistema. Un sistema de video vigilancia clásico se compone de cámaras, lente, monitores, grabadoras, matrices de video y líneas de transmisión. Con la aparición de los sistemas de video vigilancia IP, aparecen nuevos elementos, tales como grabadores digitales, servidores de video, etc. (García, 2010)

Para la investigación se toma como base de conocimiento que los sistemas de video vigilancia consisten en diversos componentes (cámara, monitor, grabadoras, etc.) conectados entre sí y controlados por un software. El mismo es el encargado de supervisar y registrar cualquier actividad física captada por las cámaras de video y solo el personal autorizado puede acceder a la información que se almacena.

#### **1.1.2 Recuperación de Información**

La Recuperación de Información (RI) consiste en la recuperación de aquellos documentos que sean relevantes y al mismo tiempo descarta los irrelevantes mediante una consulta por parte de la persona interesada. La RI en la actualidad adquiere un rol más importante debido al valor que tiene la información. Se puede plantear que disponer o no de la información justa en tiempo y forma puede resultar en el éxito o fracaso de una operación (Tolosa, y otros, 2007). De aquí, la importancia de los Sistemas de Recuperación de Información que son herramientas que permiten procesar consultas realizadas por el usuario para acceder a determinada información que se encuentra almacenada de manera eficiente.

Para implementar el proceso de RI se hace necesario tener conocimiento acerca de las técnicas de recuperación de la información que existen para identificar los más adecuados en dependencia del tipo de información. Las técnicas de recuperación de la información son las prácticas a seguir para recuperar la información según criterios de búsqueda para obtener resultados relevantes con respecto a lo que se desea encontrar (Tolosa, y otros, 2007). Entre los tipos de información que se pueden recuperar en la actualidad se encuentran: archivos de audio, imágenes, archivos de video y archivos de texto. En la investigación se centra el estudio en la recuperación de videos grabado.

### **1.1.3 Recuperación de videos grabados**

Las empresas o entidades que poseen la gestión de materiales audiovisuales como uno de sus principales misiones pueden apoyar sus sistemas con la creación de redes de almacenamiento audiovisual, protegiendo así su patrimonio histórico. En estos casos, estas redes utilizan sistemas para el almacenamiento y la recuperación de los materiales audiovisuales, permitiendo realizar una mejor gestión de los archivos audiovisuales.

Los procesos de recuperación de materiales audiovisuales pueden ser variados, siendo adaptados a los entornos donde se apliquen y las políticas organizacionales que tenga la entidad. Un ejemplo de los sistemas de almacenamiento y recuperación de audiovisuales son los sistemas de video vigilancia y las mediatecas<sup>2</sup>. La investigación centra el estudio del proceso de recuperación de videos en el sistema Xilema-Suria 2.0 y define **recuperación de videos grabados** como el proceso donde se accede a una grabación que haya sido previamente almacenada en el sistema, pudiendo utilizar varios criterios de búsqueda.

## **1.2 Evolución de los sistemas de video vigilancia**

Según (Nilsson, 2011) la vigilancia a través de grabaciones de videos existe hace muchos años. En sus inicios eran sistemas totalmente analógicos y paulatinamente se fueron digitalizando a través de la incorporación de elementos tales como cámaras IP y servidores con aplicaciones para la gestión de los videos grabados. Estas aplicaciones evolucionaron permitiendo la incorporación de numerosas funcionalidades tales como la recuperación de los videos digitales. A continuación, se detalla esta transición:

---

<sup>2</sup> Mediatecas: Sección formada con los medios de información no librarios, especialmente audiovisuales, de la biblioteca, donde se seleccionan, reúnen, ordenan y se ponen a disposición de los usuarios (CEUNET, 2007).

**Sistemas de video vigilancia analógicos usando un grabador de video (VCR, del inglés *Video Cassette Recorder*)**

Un sistema de video vigilancia analógico que utilice un VCR representa un sistema completamente analógico formado por cámaras analógicas con salida coaxial, conectadas al VCR para grabar.

El VCR utiliza cintas magnéticas para grabar y reproducir en lo posterior la señal analógica. El vídeo no se comprime y, si se graba a una velocidad de imagen completa, una cinta durará como máximo 8 horas. En sistemas mayores, se puede conectar un *quad*<sup>3</sup> o un multiplexor entre la cámara y el VCR. El *quad*/multiplexor permite grabar el vídeo procedente de varias cámaras en un solo grabador, pero con el inconveniente que tiene una menor velocidad de imagen. Para monitorizar el vídeo, es necesario un monitor analógico.

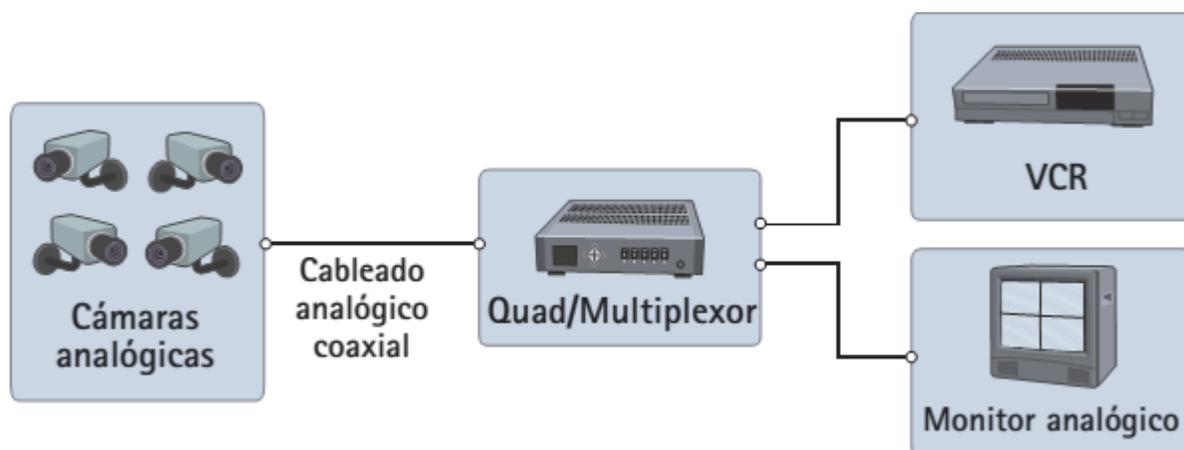


Figura 1. Sistema de video vigilancia completamente analógico. (Axis, 2014)

**Sistemas de video vigilancia analógicos usando un grabador de video digital (DVR, del inglés *Digital Video Recorder*)**

Sistema analógico con grabación digital. En un DVR, la cinta de vídeo se sustituye por discos duros para la grabación de vídeo, y es necesario que el vídeo se digitalice y comprima para almacenar la máxima cantidad de imágenes posible de un día.

Con los primeros DVR, el espacio del disco duro era limitado, por tanto, la duración de la grabación era limitada, o debía usarse una velocidad de imagen inferior. La mayoría de los DVR disponen de varias

---

<sup>3</sup> Quad: Compresor que pone en una misma pantalla hasta 4 cámaras dividiendo la pantalla en 4 cuadrantes. (SyG Seguridad, 2012)

entradas de vídeo, normalmente 4, 9 o 16, lo que significa que también incluyen la funcionalidad de los *quads* y multiplexores.

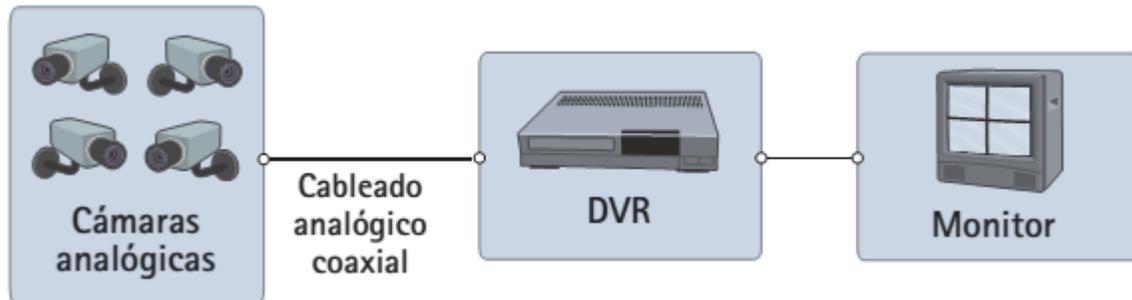


Figura 2. Sistema de video vigilancia analógico usando DVR. (Axis, 2014)

### Sistemas de video vigilancia analógicos usando un DVR de red

Sistema parcialmente digital que incluye un DVR IP equipado con un puerto Ethernet para conectividad de red. Como el vídeo se digitaliza y comprime en el DVR, se puede transmitir a través de una red informática para que se monitorice en un PC en una ubicación remota. Algunos sistemas pueden monitorizar tanto vídeo grabado como en directo, mientras otros sólo pueden monitorizar el vídeo grabado.

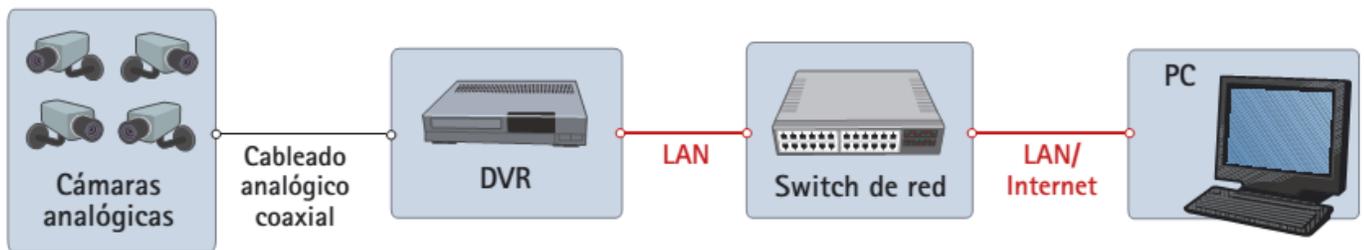


Figura 3. Sistema de video vigilancia analógico usando DVR de red. (Axis, 2014)

### Sistemas de video vigilancia IP que utilizan servidores de video

Estos sistemas de video vigilancia IP incluyen un servidor de video, un conmutador de red y una computadora con software de gestión de video. La cámara analógica se conecta al servidor de video, el cual digitaliza y comprime el video. A continuación, el servidor de video se conecta a una red y transmite el video a través de un conmutador de red a un PC, donde se almacena en discos duros.

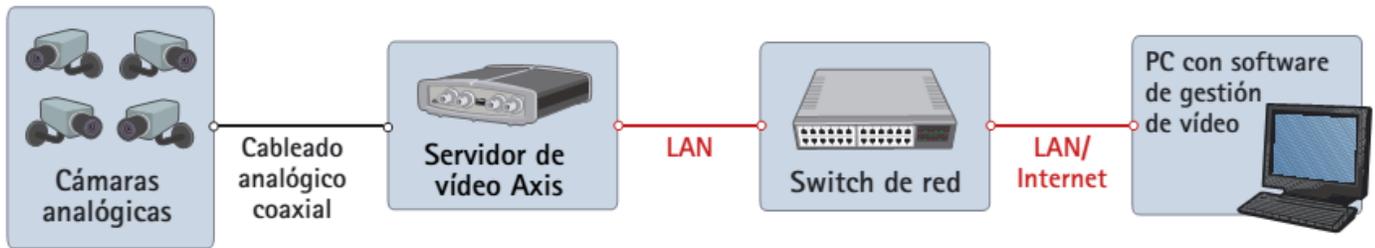


Figura 4. Sistema de video vigilancia usando servidor de video Axis.

### Sistemas de video vigilancia IP que utilizan cámaras IP

Una cámara IP combina una cámara y un ordenador en una unidad, lo que incluye la digitalización y la compresión del video, así como un conector de red. El video se transmite a través de una red IP, mediante los conmutadores de red y se graba en una computadora estándar con software de gestión de vídeo. Esto representa un verdadero sistema de video IP donde no se utilizan componentes analógicos.

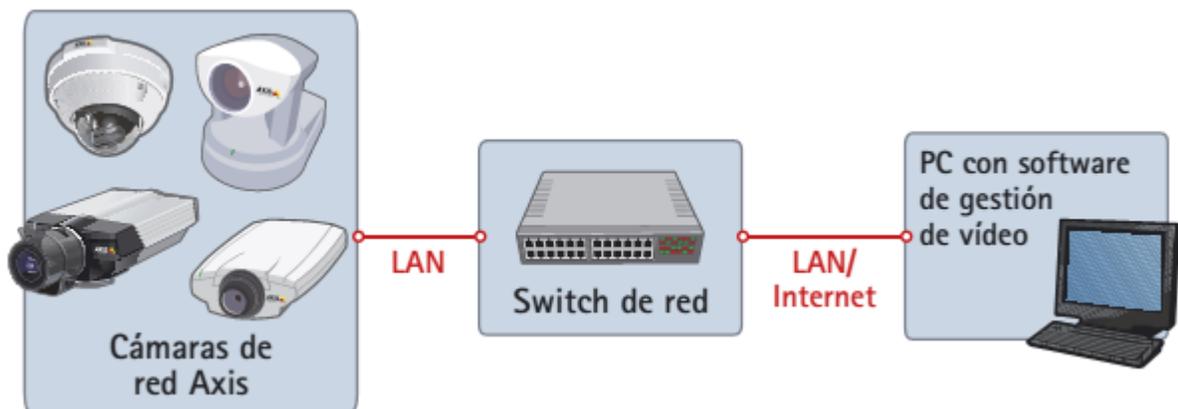


Figura 5. Sistema de video vigilancia usando cámaras de red Axis. (Axis, 2014)

## 1.3 Evolución del proceso de recuperación de grabaciones

Los sistemas de video vigilancia han evolucionado siendo cada vez más complejos y modulares, permitiendo la incorporación de numerosas funcionalidades tales como la recuperación de los videos digitales. En las últimas décadas, las técnicas de recuperación de medias en los sistemas de vigilancia, han evolucionado en cuanto al tipo de tecnología utilizada. En el acápite anterior se explica la evolución de estos sistemas en cinco tipos diferentes, para la descripción de la evolución que ha tenido el proceso de recuperación de grabaciones se agrupan los tipos de sistemas en tres generaciones:

Tabla 1. Evolución de los sistemas de video vigilancia

Generación	Tipos de sistemas de video vigilancia	
Totalmente analógicos	Sistemas de video vigilancia analógicos usando VCR.	
Parcialmente digitales	Sistemas de video vigilancia analógicos usando DVR.	
	Sistemas de video vigilancia analógicos usando DVR de red.	
Totalmente digitales	Sistemas de video vigilancia IP usando servidores de video.	
	Sistemas de video vigilancia IP usando servidores y cámaras IP.	

En la primera generación el proceso de recuperación de videos se realizaba de forma manual, por lo que se hacía bien complejo el trabajo con las mismas ya que podía llegar a existir una gran cantidad de cintas grabadas y cuando se deseaba localizar algún hecho en específico había que revisar cada una de ellas hasta encontrar la secuencia de video deseada.

En la segunda generación, el empleo de los dispositivos de almacenamiento, en vez de los *cassettes*, permitió el almacenamiento de las transmisiones digitales sin la necesidad de la intervención humana o cambio de cintas, aumentando así los tiempos de grabación y permitiendo acoger simultáneamente varias secuencias de video. Aunque el proceso de recuperación aún se realizaba de forma manual, pues el encargado debía buscar la grabación para luego visualizarla hasta encontrar la secuencia o el evento deseado.

Finalmente, en la tercera generación ya se incorporan los softwares de gestión de video con la capacidad de realizar cada una de las tareas que se realizaban de forma manual ya que se crearon módulos para garantizar la búsqueda inteligente de grabaciones en la base de datos de la aplicación y la visualización de los videos digitales guardados sin importar la cantidad de cámaras o archivos que se encuentren almacenados en el servidor.

## 1.4 Descripción de la situación problemática

El centro GEYSED de la UCI posee el Sistema de Video Vigilancia Xilema-Suria 2.0 que cuenta con una arquitectura que le permite adaptarse a los modelos de cámaras mediante el desarrollo de *plugin* para la integración con los distintos modelos de cámaras. Este sistema tiene un video sensor integrado que detecta movimiento por cámaras y emite alertas sonoras y visuales en función de la alarma activada. El sistema cuenta igualmente con seis módulos:

**Módulo Gestor:** Es el módulo fundamental de la aplicación, el resto de los módulos funcionan como 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.

**Módulo de 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.

**Módulo Visor:** Es el módulo que permite visualizar los flujos de videos capturados por las cámaras. Pueden existir varias instancias de esta estación corriendo en los dominios físicos del sistema. 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. También permite manipularlas en la medida de las capacidades de cada una.

**Módulo de Análisis:** Es el módulo encargado de realizar el procesamiento a los flujos de videos que le sean asignados, ya sea por planificación o bajo demanda. El mismo posee los videos sensores que permiten procesar los datos que emiten las cámaras.

**Módulo de Grabación:** Es el módulo encargado de realizar las grabaciones, ya sea por planificación o bajo demanda.

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

El proceso de recuperación de videos grabados en Xilema-Suria 2.0 se realiza accediendo a los directorios de almacenamiento físico, donde se encuentran las grabaciones realizadas de forma diaria. Este proceso resulta engorroso debido a que no existe una herramienta o módulo encargado de recuperar las grabaciones realizadas y almacenadas en el sistema, lo que trae como consecuencia:

- ✓ El acceso indebido a materiales restringidos ya que para acceder a las grabaciones es mediante los exploradores nativos del sistema operativo donde se encuentre instalado el módulo de autonomía. Por lo tanto, compromete la integridad de los datos.
- ✓ El acceso no controlado a las grabaciones propicia la corrupción de los mismos, ya que los operadores eliminan los archivos de manera no controlada, incidiendo negativamente en la salva guarda de los medios protegidos por el sistema y creando situaciones de conflictos en cuanto a la irresponsabilidad probada en función del solapamiento de sucesos de interés.
- ✓ Se generan grandes volúmenes de información en cuanto a las grabaciones ya que las cámaras graban las 24 horas del día en cortes de cada media hora. Provocando que la recuperación de estos archivos en los dispositivos de almacenamiento tienda a demorarse e incluso que no siempre se obtenga la información de manera oportuna.
- ✓ Dificultad para identificar rápidamente los materiales de interés ya que el nombre de los archivos de grabación posee una nomenclatura específica generada por el módulo de grabación.
- ✓ Su visualización y análisis se realiza empleando los reproductores de video del propio sistema operativo, externos al sistema Xilema-Suria 2.0, los cuales no realizan la extracción de segmentos de interés de la grabación para luego ser tipificados y utilizados para integrar información útil al proceso de protección con el sistema.

## 1.5 Revisión de sistemas nacionales e internacionales

Para lograr una mejor comprensión de las características del módulo a desarrollar, se hace necesario realizar un análisis de algunos sistemas de video vigilancia que realicen la recuperación de videos grabados.

### 1.5.1 Sistema de video vigilancia *AXIS Camera Station*

AXIS es una empresa que ofrece soluciones de software de vigilancia IP como complemento a las cámaras de red y los codificadores de video de la empresa. Una de las soluciones de sistemas de video vigilancia que ha desarrollado esta empresa es el *AXIS Camera Station*. El sistema incorpora características de seguridad tales como: múltiples niveles de acceso de usuario con protección por contraseña y una base de datos con grabaciones ilimitadas, solo restringidas por el espacio del disco. Además, es capaz de proporcionar funciones de supervisión de video, grabación y gestión de eventos, cuenta con múltiples funciones de búsqueda de grabaciones, por fecha, hora o movimiento, también incorpora una potente función de búsqueda con visualización cronológica de grabaciones y capacidades de exportación. La compatibilidad con planos y mapas permite captar una visión panorámica de la zona supervisada, así como obtener un acceso rápido a cualquier cámara IP que haya en la instalación. (Axis, 2014)

### **1.5.2 Sistema de video vigilancia SavVi**

SavVi es una solución unificada de software de análisis informático de video que ofrece una amplia variedad de funciones analíticas a través de una plataforma singular y fácil de usar que se integra a sistemas de vigilancia. Algunas de las funcionalidades que ofrece son: la detección de eventos en tiempo real, la búsqueda de video y aplicaciones de inteligencia de negocios. Realiza una automática y cómoda recuperación y análisis de videos grabados que reemplaza las búsquedas manuales incómodas y trabajosas. Los usuarios definen parámetros respecto al suceso/objeto de interés y reciben resultados de búsqueda concordantes en segundos, permitiendo el acceso rápido a segmentos de video específicos enterrados en el video almacenado.

### **1.5.3 Sistema de video vigilancia Xyma Safe Vision**

La empresa de desarrollo de tecnologías y sistemas DATYS cuenta con un Sistema de Video Protección profesional basado en tecnología IP llamado *Xyma Safe Vision*. Permite administrar, monitorear, grabar, revisar, configurar alertas y alarmas, así como obtener reportes, para cualquier solución de video vigilancia que se requiera implementar. Está compuesto por 8 tipos de módulos: Administrador, Monitor, *Analytics*, Alertas, Grabaciones, Revisor, Reportes y Web. Los módulos interactúan entre sí y comparten una base de datos y pueden estar concentrados en una PC o distribuidos tantas veces y en tantas máquinas como requiera la extensión del sistema. Este software nacional es un gran avance en el desarrollo de este tipo de soluciones en Cuba. (Reinoso Medina, y otros, 2016)

### **1.5.4 Sistema de video vigilancia de la Dirección de Comunicación Institucional UCI**

La Dirección de Comunicación Institucional (DCI) de la UCI tiene un sistema de video vigilancia analógico, marca Panasonic. El mismo posee un visor por cada cámara de las que cuenta el sistema. Utilizan un *switch* para multiplexar los flujos de las cámaras y mostrarlas en un solo monitor. Permite grabar flujos de cámara, configurar las cámaras, visualizar los flujos entre otras acciones. El sistema solo almacena las grabaciones donde ocurre algo inusual y cuando el operador desea acceder a una grabación almacenada debe hacerlo accediendo directamente al servidor, debido a que el sistema no cuenta con la funcionalidad de recuperación.

### **1.5.5 Análisis de las soluciones similares.**

A continuación, se muestra una tabla resumen de las principales características que se tomaron en cuenta para el estudio de las soluciones similares. Las características a evaluar son:

**Búsqueda de grabaciones:** Se analiza de las soluciones encontradas si realizan la búsqueda por filtros o no.

**Reproductor incluido:** Se analiza si poseen un reproductor incluido en el propio sistema.

**Control de acceso:** Se analiza si se realiza un control de acceso al sistema.

**Registro de actividades:** Se analiza si los sistemas realizan un control y registro de las actividades realizadas por el usuario.

**Integración con Xilema-Suria:** Se analiza si los sistemas tienen una interfaz de integración.

Tabla 2. Características de las soluciones analizadas

Características Sistemas	Búsqueda de grabaciones	Reproductor incluido	Control de acceso	Registro de actividades	Integración con Xilema-Suria
AXIS Camera Station	-	X	-	-	-
SavVi	X	X	X	-	-
Xyma Safe Vision	-	X	-	X	-
Sistema de video vigilancia DCI de la UCI.	-	X	-	X	-

Luego de finalizar la investigación a nivel internacional y nacional sobre el proceso de recuperación de videos grabados en sistemas de video vigilancia, se puede concluir que todos tienen características que sirven de base para la conceptualización de una solución para la problemática de la investigación. Igualmente, se descartan todas las aplicaciones revisadas por no poseer una interfaz o procesos de integración que permitan al sistema Xilema-Suria 2.0 contar con sus funcionalidades en el proceso de recuperación de sus materiales audiovisuales grabados.

## 1.6 Metodología, herramientas y tecnologías a utilizar

Para desarrollar el módulo recuperador propuesto en la presente investigación y lograr una fácil integración con el sistema se emplearán las herramientas, tecnologías y metodología que se definieron en el proyecto. A continuación, se procede con la descripción de las mismas.

### 1.6.1 Metodología de desarrollo de software: AUP-UCI

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. Surge ante la necesidad de trabajar mediante el uso de procedimientos, técnicas, herramientas y documentos durante el desarrollo de software. (Pressman, 2010)

El Proceso Unificado Ágil (AUP, del inglés *Agile Unified Process*) de Scott Ambler es una versión simplificada de la metodología Proceso Unificado de *Rational* (RUP, del inglés *Relational Unified Process*). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. (UCI, 2015)

En el proyecto de video vigilancia la metodología que se utiliza es una variación realizada en la UCI de “Proceso Unificado Ágil”. Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto, se utiliza la metodología de desarrollo AUP-UCI. La misma define que para el ciclo de vida de los proyectos de la UCI, de las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición), mantener la fase de Inicio, las 3 fases restantes de AUP se agrupan en una sola, que es Ejecución y se agrega la fase de Cierre (UCI, 2015). Conjuntamente especifica principios, etapas o flujos de trabajo, para mejorar y organizar el proceso de desarrollo desde su comienzo. Además, con dicha metodología se logra estandarizar el proceso de desarrollo de software y hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos.

### 1.6.2 Lenguaje de modelado: UML

Un lenguaje de modelado es utilizado en combinación con una metodología de desarrollo de software para llegar de una especificación inicial a la implementación (Pressman, 2010). Para el desarrollo del sistema se hará uso del Lenguaje de Modelado Unificado (UML, del inglés *Unified Modeling Language*) en su versión 2.1 por ser el estándar más utilizado para especificar y documentar cualquier sistema de forma precisa. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software (Jacobson, y otros, 2000).

UML está diseñado para ser un lenguaje de modelado de propósito general proporcionando una gran flexibilidad y expresividad a la hora de modelar sistemas. Puede ser utilizado para especificar la mayoría de los sistemas basados en objetos o en componentes y para modelar aplicaciones de muy diversos dominios de aplicación y plataformas de objetos distribuido. (Larman, 2005)

### 1.6.3 Herramienta CASE: *Visual Paradigm for UML*

Una herramienta CASE (del inglés *Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora), es un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases. (Case, 2001)

*Visual Paradigm* es una herramienta CASE que utiliza UML como lenguaje de modelado, soporta el ciclo de vida completo del proceso de desarrollo de software, a través de la representación de todo tipo de

diagramas. Se utiliza *Visual Paradigm for UML* en su versión 8.0 ya que usa un lenguaje estándar común para todo el equipo de proyecto, además es compatible con diferentes lenguajes, entre ellos C++. También permite realizar la construcción de diferentes modelos (despliegue, componente, clases, entre otros) además de aportar una documentación detallada de los mismos.

#### 1.6.4 Lenguaje de Programación: C++

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura, así como el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina. (Microsoft Corporation, 2011)

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 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. (C++, 2014)

Por lo antes expuesto y para lograr la integración con el sistema Xilema-Suria 2.0 se utilizará C++ en su versión 11 ya que permite incorporar librerías, según la necesidad del desarrollo de aplicaciones y gracias a su capacidad de uso de objetos y clases, además, facilita la reutilización del código.

#### 1.6.5 Framework de desarrollo: Qt

Un marco de trabajo (*framework*) en el desarrollo de software, es una estructura de soporte definida mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. (Codebox, 2010) Qt es un marco de trabajo multiplataforma, que se utiliza para el desarrollo de aplicaciones, está escrito en C++, sin embargo, es posible utilizar Qt con otros lenguajes. Funciona en todas las principales plataformas, y tiene un amplio apoyo. Es un software libre y de código abierto. (QT Centre, 2014)

Se selecciona Qt en su versión 5.2 como marco de trabajo para el desarrollo de la interfaz gráfica del módulo por lo antes expuesto y debido a que proporciona un grupo de elementos gráficos predefinidos y lo necesario para construir interfaces de usuario modernas.

#### 1.6.6 Entorno de Desarrollo Integrado: Qt Creator

Un Entorno de Desarrollo Integrado (IDE, del inglés *Integrated Development Environment*) 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#, entre otros, logrando utilizarse en el mismo uno o varios lenguajes de programación. (Fritzler, 2011)

Para el desarrollo del Módulo Recuperador para Xilema-Suria 2.0 se utilizará Qt *Creator* en su versión 3.0 por ser un potente IDE multiplataforma, diseñado a la medida de los desarrolladores de Qt, para hacer que el desarrollo en C++ de las aplicaciones Qt sea más rápido y fácil. Permite a los desarrolladores crear aplicaciones para escritorio. Además, posee un avanzado editor de código C++, lo cual facilita la programación de la aplicación ofreciendo completado de código.

### 1.6.7 Biblioteca: ZMQ

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.

ZMQ es una tecnología de comunicación que permite el intercambio de mensajes a través de comunicación TCP/IP de forma rápida y sencilla (Hintjens, 2002). Para la solución propuesta se utilizará la tecnología de comunicación ZMQ, ya que es la determinada por el sistema Xilema-Suria 2.0 para establecer la comunicación entre sus módulos. Además, los sistemas de comunicación diseñados con esta biblioteca ofrecen rendimiento, robustez y compatibilidad con C++.

Dicha biblioteca fue modificada por los desarrolladores del proyecto, donde implementaron sobre la capa de comunicación, específica para el envío y recepción de la información en todo el sistema. Esto propicia una manera particular y adaptada de compartir la información entre los módulos, lo que aporta a la distribución segura de los datos.

## 1.7 Conclusiones del capítulo

Con el estudio de los elementos presentes en este capítulo, se les da cumplimiento a las tareas de la investigación 1, 2 y 3, lo que se evidencia en:

- ✓ El estudio de los conceptos asociados a la investigación y la evolución de los sistemas de video vigilancia, permitió obtener un mayor dominio del problema a resolver.
- ✓ El análisis de soluciones existentes tanto a nivel nacional como internacional, demostró que estas no satisfacen las necesidades de Xilema-Suria 2.0 debido a 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.

- 
- ✓ La caracterización de la metodología, las tecnologías y herramientas seleccionadas para la construcción y desarrollo del módulo propuesto, permitió conocer las ventajas que ofrecen las mismas para el desarrollo del módulo y lograr la integración con el Sistema de Video Vigilancia Xilema-Suria 2.0.

## Capítulo 2. “Características del Módulo Recuperador para Xilema-Suria 2.0”

En el presente capítulo se describe la solución propuesta, a partir de realizar un estudio inicial del entorno en el cual se enmarca el problema a resolver. También se identifican y describen los requisitos funcionales y no funcionales, agrupándose en casos de uso y brindando información referente a los mismos y a los actores del sistema, de manera que se pueda entender mejor el producto que se desea desarrollar. Se definen además los patrones de diseño y arquitectónicos empleados. Por último, se muestra el diagrama de clases del diseño propuesto.

### 2.1 Modelo de dominio

Se realiza el modelo de dominio debido a que no existe un negocio definido, por lo cual, no se pueden determinar los procesos y roles del proceso de negocio, haciéndose engorroso y poco exacto la descripción de los mismos. Además, no se cuenta con un cliente, por lo que se determina que no es viable llevar a cabo un modelo de negocio y se hace necesario realizar la descripción del proceso de recuperación mediante un modelo de dominio.

Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan los conceptos que existen o los eventos que suceden en el entorno en el que trabaja el sistema. (Sommerville, 2005)

#### 2.1.1 Diagrama del modelo de dominio

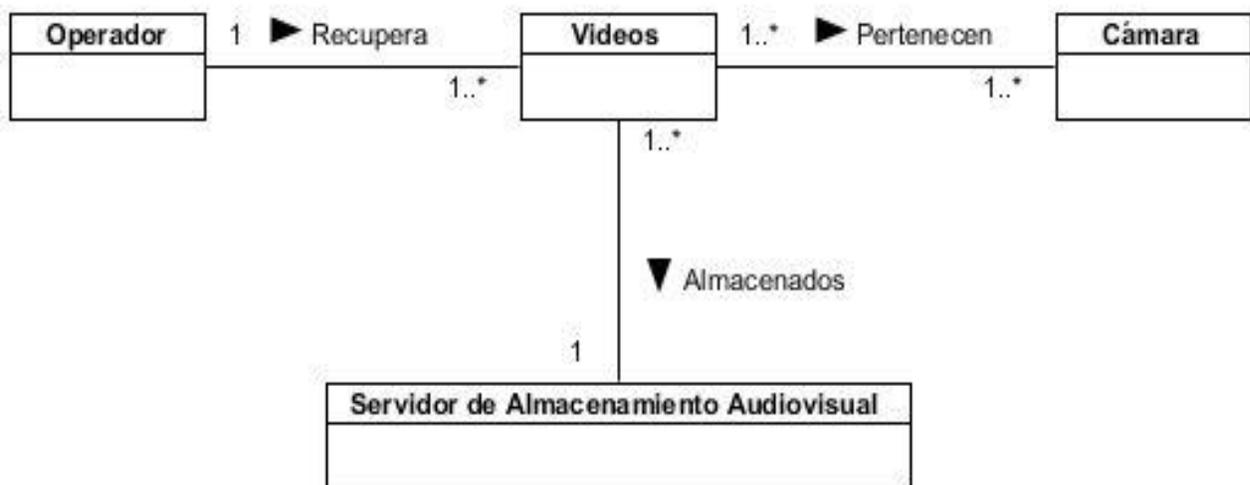


Figura 6. Modelo del dominio del Módulo Recuperador para Xilema-Suria 2.0

### 2.1.2 Descripción de los conceptos que intervienen en el dominio del problema

**Operador:** Persona encargada de la recuperación de videos.

**Videos:** Serie de imágenes que pertenecen a una cámara y son almacenados en el servidor de administración.

**Cámara:** Dispositivo diseñado para enviar los flujos de video obtenidos de un área en específico a través de la red.

**Servidor de almacenamiento audiovisual:** Servidor donde se almacenan los videos capturados por las cámaras.

### 2.1.3 Descripción del flujo del modelo de dominio

Actualmente el proceso de recuperación de videos almacenados en Xilema-Suria 2.0 se realiza manualmente. El operador es el encargado de realizar la recuperación de los videos grabados. Cada grabación pertenece a una cámara y son almacenados en el servidor de almacenamiento audiovisual.

## 2.2 Especificación de los requisitos

La especificación de requisitos en el proceso de desarrollo del software es de vital importancia. Tener los requisitos bien claros y definidos permite comprender desde un inicio la línea a seguir en el desarrollo y así garantizar la eficiencia y calidad del software (Pressman, 2010). La misma incluye los requisitos funcionales, que son condiciones que el sistema debe cumplir y los requisitos no funcionales, que son cualidades que el producto debe tener.

### 2.2.1 Requisitos funcionales

Los requisitos funcionales son declaraciones de servicios que el sistema debe proporcionar, define la manera en que éste debe reaccionar a determinadas entradas y cómo se debe comportar en situaciones particulares (Pressman, 2010). A través de la entrevista realizada a los miembros del proyecto de video vigilancia se obtienen los requisitos funcionales (RF) que el Módulo Recuperador para Xilema-Suria 2.0 debe cumplir.

#### **RF1. Autenticar usuario.**

**Descripción:** Permite a los usuarios acceder al módulo, otorgándole los permisos y privilegios que le correspondan según el rol que tenga en el sistema, garantizando la seguridad de acceso al módulo.

#### **RF2. Cerrar sesión.**

**Descripción:** Permite a los usuarios previamente autenticados salir del módulo.

**RF3. Realizar búsqueda de videos grabados por zona.**

**Descripción:** Permite a los usuarios con permisos de recuperador, realizar búsquedas de videos grabados por zona.

**RF4. Realizar búsqueda de videos grabados por cámara.**

**Descripción:** Permite a los usuarios con permisos de recuperador, realizar búsquedas de videos grabados por cámara.

**RF5. Realizar búsqueda de videos grabados por rango de fecha y hora.**

**Descripción:** Permite a los usuarios con permisos de recuperador, realizar búsquedas de videos grabados definiendo un rango de fecha y hora.

**RF6. Realizar búsqueda de videos grabados por evento.**

**Descripción:** Permite a los usuarios con permisos de recuperador, realizar búsquedas de videos grabados por eventos.

**RF7. Reproducir video grabado.**

**Descripción:** Permite reproducir el video seleccionado.

**RF8. Pausar video grabado.**

**Descripción:** Permite pausar el video seleccionado.

**RF9. Detener video grabado.**

**Descripción:** Permite detener el video seleccionado.

**RF10. Control de volumen de reproducción.**

**Descripción:** Permite subir o disminuir el volumen del reproductor.

**RF11. Control de velocidad de reproducción.**

**Descripción:** Permite visualizar el video seleccionado a una menor o mayor velocidad.

**RF12. Control de la reproducción.**

**Descripción:** Permite retroceder o adelantar el video seleccionado a un determinado instante de tiempo.

**RF13. Establecer vista del espacio de trabajo.**

**Descripción:** Establece una vista del espacio de trabajo según desee el recuperador.

**RF14. Mostrar metadatos de la cámara.**

**Descripción:** Permite que el usuario al seleccionar la cámara deseada conozca los metadatos correspondientes a dicha cámara, ya sea: resolución, modelo, existencia de zoom y el tipo de rotación.

**RF15. Mostrar metadatos del video grabado.**

**Descripción:** Muestra al usuario los metadatos correspondientes al video que desee, ya sea: hora de inicio y fin, duración, zona y cámara.

**RF16. Generar árbol de cámaras.**

**Descripción:** Permite generar el árbol de cámara, una vez que el usuario entra al módulo, con los videos del día anterior.

**RF17. Registrar trazas de recuperador.**

**Descripción:** Permite realizar un registro de las trazas de cada usuario que interactúe con el módulo.

**RF18. Mostrar Trazas de recuperador.**

**Descripción:** Permite mostrar las trazas de cada usuario previamente registradas por el módulo.

**RF19. Configurar conexión con el gestor desde el recuperador.**

**Descripción:** Permite configurar la conexión con el Gestor desde el recuperador, especificando la dirección IP donde se encuentra el Gestor y el puerto por el cual se realizará la comunicación con el Gestor.

**RF20. Editar configuración.**

**Descripción:** Permite editar la configuración previamente especificada.

### **2.2.2 Requisitos no funcionales**

Los requisitos no funcionales (RNF) detallan las propiedades o cualidades que el producto debe tener, aumentándole funcionalidad al sistema, pues hacen al producto atractivo, fácil de usar, rápido y confiable (Pressman, 2010).

**Funcionalidad:**

**Seguridad:** El sistema deberá permitir la protección de la información y datos de manera que los usuarios no autorizados no puedan leerlos o modificarlos, al tiempo que no se deniega el acceso a los usuarios autorizados.

**Usabilidad:**

Tipo de Aplicación Informática: El Módulo Recuperador para Xilema-Suria 2.0 será una aplicación de escritorio.

**Requisitos de Software:**

La computadora donde se instale el módulo debe tener instalada la biblioteca ZMQ en su versión 4.0.3 y sistema operativo (SO) Linux, distribución de Ubuntu 15.10.

**Requisitos de Hardware:**

Están dados por las prestaciones mínimas necesarias para el trabajo de la Herramienta de Análisis de Videos. La misma utiliza OpenCV en su versión 3.0, para una arquitectura de 64 bits donde se encuentran implementados y optimizados los procedimientos que son usados en el trabajo con los videos. Dicha herramienta debe de coexistir como parte del módulo que se desarrolla en esta investigación, por lo que los requisitos deben de soportar ambas soluciones. La computadora donde se instale el módulo debe tener como mínimo:

- Procesador: Core 2 Duo a 1.6 GHz.
- Memoria RAM: 1 GB.
- Red LAN interna a 100 MB.

**Restricciones de Integración:**

Para realizar la integración entre el módulo recuperador de videos y el sistema Xilema-Suria 2.0 se debe utilizar la biblioteca ZMQ en su versión 4.0.3.

## **2.3 Modelo de casos de uso del sistema**

El modelo de casos de uso captura todos los requisitos funcionales del módulo, permitiendo a los desarrolladores de software, usuarios e interesados entender el funcionamiento del mismo. Es un modelo del módulo el cual contiene a los actores y casos de uso identificados, así como sus relaciones. Estos casos de uso, actores y relaciones son representados en un diagrama de casos de uso el cual modela el comportamiento del módulo.

### **2.3.1 Casos de uso del sistema**

Los casos de uso (CU) son un conjunto de escenarios que identifican una línea de utilización para el sistema que va a ser desarrollado, facilitando una descripción de cómo se usará el sistema (Pressman, 2010). Los requisitos funcionales definidos anteriormente se agruparon en los siguientes casos de uso:

**CU 1: Autenticar usuario.**

RF1. Autenticar usuario.

**CU 2: Cerrar sesión.**

RF2. Cerrar sesión.

**CU 3: Realizar búsquedas.**

RF3. Realizar búsqueda de videos grabados por zona.

RF4. Realizar búsqueda de videos grabados por cámara.

RF5. Realizar búsqueda de videos grabados por rango de fecha y hora.

RF6. Realizar búsqueda de videos grabados por evento.

**CU 4: Realizar operaciones en el reproductor.**

RF7. Reproducir video grabado.

RF8. Pausar video grabado.

RF9. Detener video grabado.

RF10. Control de volumen de reproducción.

RF11. Control de velocidad de reproducción.

RF12. Control de la reproducción.

**CU 5: Establecer vista del espacio de trabajo.**

RF13. Establecer vista espacio de trabajo.

**CU 6: Mostrar metadatos.**

RF14. Mostrar metadatos de la cámara.

RF15. Mostrar metadatos del video grabado.

**CU 7: Generar árbol de cámaras.**

RF16. Generar árbol de cámaras.

**CU 8: Registrar trazas.**

RF17. Registrar trazas de recuperador.

**CU 9: Mostrar trazas.**

RF18. Mostrar Trazas de recuperador.

**CU 10: Administrar conexión con el gestor.**

RF19. Configurar conexión con el gestor desde el recuperador.

RF20. Editar configuración.

**2.3.2 Descripción del actor del sistema**

Los actores representan personas, sistemas o hardware externo que interactúan con el sistema. A continuación, se muestra una tabla con la descripción del actor que interactúa con el módulo.

Tabla 3. Actor del sistema

Actor	Descripción
Recuperador	Representa los usuarios que tienen permiso para realizar las funciones del recuperador como son: realizar búsquedas, reproducir los videos grabados, configurar la conexión con el gestor, entre otras, sin tener acceso a la edición de los videos grabados.

**2.3.3 Diagrama de caso de uso del sistema**

Un diagrama de caso de uso muestra la relación entre los actores del sistema y los casos de uso (Pressman, 2010). La siguiente figura muestra el diagrama de casos de uso del Módulo Recuperador para Xilema-Suria 2.0.

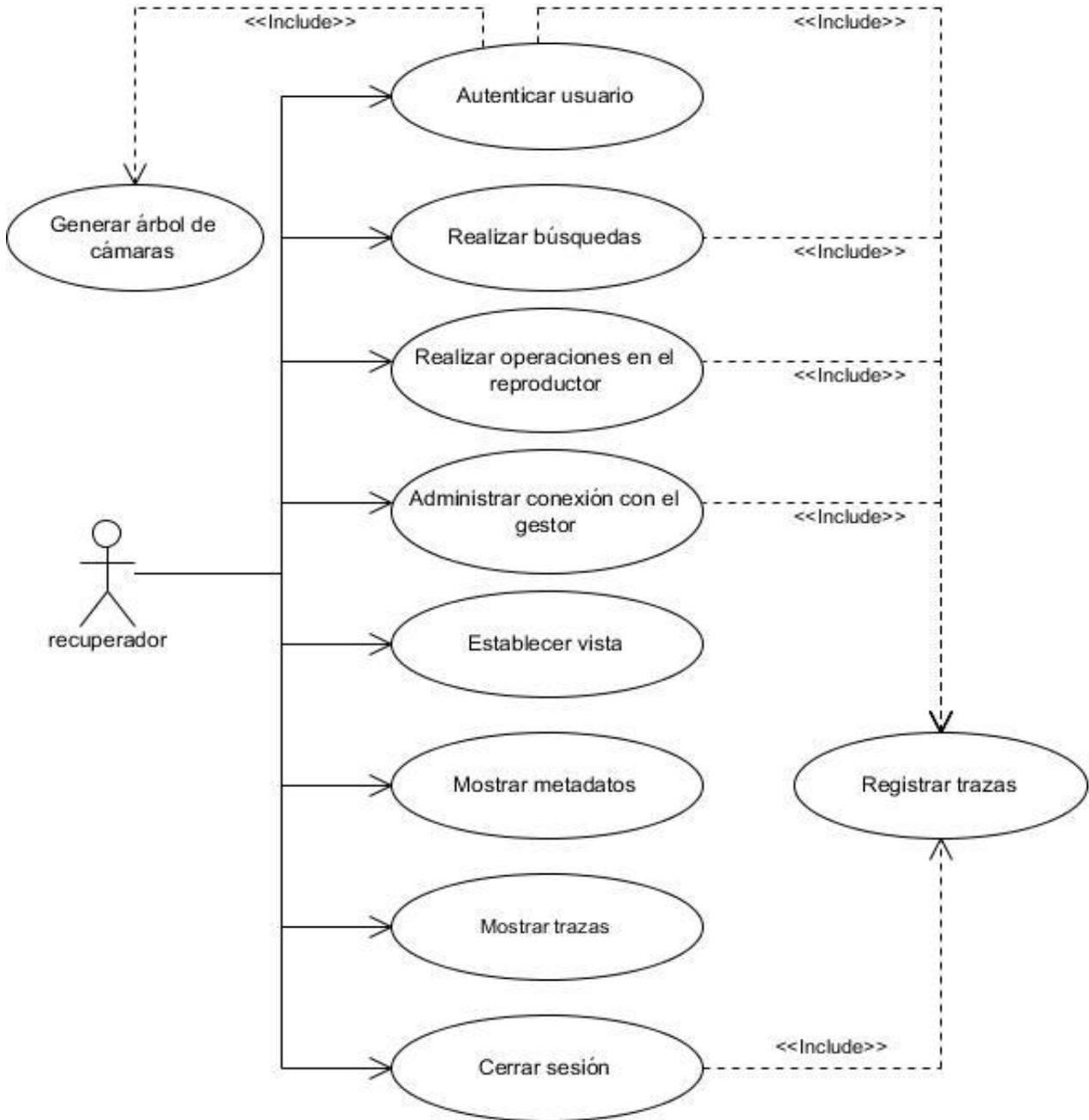


Figura 7. Diagrama de casos de uso del Módulo Recuperador para Xilema-Suria 2.0

### 2.3.4 Descripción de los casos de uso del sistema

La descripción amplia de los casos de uso del sistema permite realizar el diseño, implementación y las pruebas de la interfaz. A continuación, se describe el caso de uso Realizar búsqueda.

Nota: La descripción de los otros casos de uso están en los anexos.

Tabla 4. Descripción del CU: Realizar Búsquedas

<b>Objetivo</b>	Realizar búsquedas de las grabaciones almacenadas por el sistema facilitando localizar grabaciones específicas.	
<b>Actores</b>	Recuperador: (Inicia) Búsqueda de grabaciones, por fecha, hora, eventos, zonas y cámaras.	
<b>Resumen</b>	El CU inicia cuando se solicita realizar búsquedas de grabaciones y termina cuando se le muestra el listado de grabaciones.	
<b>Complejidad</b>	Media.	
<b>Prioridad</b>	Crítico.	
<b>Precondiciones</b>	Módulo Recuperador conectado al Gestor. Usuario autenticado en el sistema.	
<b>Postcondiciones</b>	El usuario obtiene el video grabado o los videos grabados que estaba buscando.	
<b>Referencia</b>	<b>RF3.</b> Realizar búsqueda de videos grabados por zona. <b>RF4.</b> Realizar búsqueda de videos grabados por cámara. <b>RF5.</b> Realizar búsqueda de videos grabados por rango de fecha y hora. <b>RF6.</b> Realizar búsqueda de videos grabados por evento.	
<b>Flujo de eventos</b>		
<b>Flujo básico: Realizar búsquedas de grabaciones.</b>		
	<b>Actor</b>	<b>Sistema</b>
1	Selecciona la opción de realizar búsqueda de grabaciones.(Ver Prototipo 1)	
2		Muestra una interfaz con las siguientes opciones (ver Prototipo 2): <ul style="list-style-type: none"> <li>- Buscar por zona (ver Sección 1: Buscar por zona). (A)</li> <li>- Buscar por cámaras (ver Sección 2: Buscar por cámaras). (B)</li> <li>- Buscar por rango de fecha (ver Sección 3: Buscar por rango de fecha). (C)</li> <li>- Buscar por rango de hora (ver Sección 4: Buscar por rango de hora). (D)</li> <li>- Buscar por eventos. (E)</li> </ul> <b>Nota:</b> Cuando el campo de zonas es marcado, el campo de las cámaras se modifica mostrando solamente las

		<p>cámaras correspondientes a la zona seleccionada.</p> <p>Los eventos son los detectados por el video sensor y los definidos por el actor.</p>
3	Selecciona el criterio que desea para buscar las grabaciones.	
4	Selecciona la opción "Buscar".	
5		Solicita al Gestor los datos de las grabaciones, teniendo en cuenta los parámetros seleccionados (fecha, hora, zona y cámara).
6		<p>Muestra la interfaz del Recuperador actualizando el árbol de grabaciones con los videos correspondientes a la búsqueda realizada.</p> <p>Termina el CU.</p>
<b>Flujos alternos</b>		
<b>Nº Evento 3a. El usuario no selecciona ningún criterio de búsqueda.</b>		
	<b>Actor</b>	<b>Sistema</b>
3a		Muestra el árbol de cámaras con todas las grabaciones almacenadas del día anterior.
<b>Nº Evento 4a. El usuario selecciona la opción "Cancelar".</b>		
	<b>Actor</b>	<b>Sistema</b>
4a		<p>Cancela la operación.</p> <p>Termina el CU</p>
<b>Sección 1: Buscar por zona.</b>		
<b>Flujo básico: Selecciona la opción de buscar por zona.</b>		
	<b>Actor</b>	<b>Sistema</b>

1		Muestra un listado con las zonas existentes en el sistema.
2	Selecciona la zona.	
3		Busca en el servidor de almacenamiento las grabaciones correspondientes a las cámaras según la zona seleccionada.

**Sección 2: Buscar por cámara.**

**Flujo básico: Selecciona la opción de buscar por cámara.**

	Actor	Sistema
1		Muestra un listado con las cámaras existentes en el sistema.
2	Selecciona la cámara que desea.	
3		Busca en el servidor de almacenamiento las grabaciones correspondientes a la cámara seleccionada.

**Sección 3: Buscar por rango de fecha.**

**Flujo básico: Selecciona la opción de buscar por rango de fecha.**

	Actor	Sistema
1		Muestra dos campos para la fecha inicial y la fecha final.
2	Establece el rango de fecha que desee.	
3		Busca en el servidor de almacenamiento las grabaciones realizadas en el rango de fechas establecido.

**Flujos alternos**

**Nº Evento 2. El usuario establece el rango de fecha incorrectamente.**

	Actor	Sistema
--	-------	---------

2.1		Muestra un mensaje alertando al usuario del error.
2.2	Selecciona la opción de "Aceptar"	
2.3		Vuelve a la interfaz del Buscar.
<b>Sección 4: Buscar por rango de hora.</b>		
<b>Flujo básico: Selecciona la opción de buscar por rango de hora.</b>		
	<b>Actor</b>	<b>Sistema</b>
1		Muestra dos campos para la hora inicial y la hora final.
2	Establece el rango de hora que desee.	
3		Busca en el servidor de almacenamiento las grabaciones realizadas en el horario definido por el actor.
<b>Flujos alternos</b>		
<b>Nº Evento 2. El usuario establece el rango de hora incorrectamente.</b>		
	<b>Actor</b>	<b>Sistema</b>
2.1		Muestra un mensaje alertando al usuario del error.
2.2	Selecciona la opción de "Aceptar".	
2.3		Vuelve a la interfaz del Buscar.
<b>Sección 4: Buscar por evento.</b>		
<b>Flujo básico: Selecciona la opción de buscar por evento.</b>		
	<b>Actor</b>	<b>Sistema</b>
1		Busca en el servidor de almacenamiento las grabaciones donde ocurrió algún evento.
<b>Relaciones</b>	CU Incluidos	No procede
	CU Extendidos	No procede

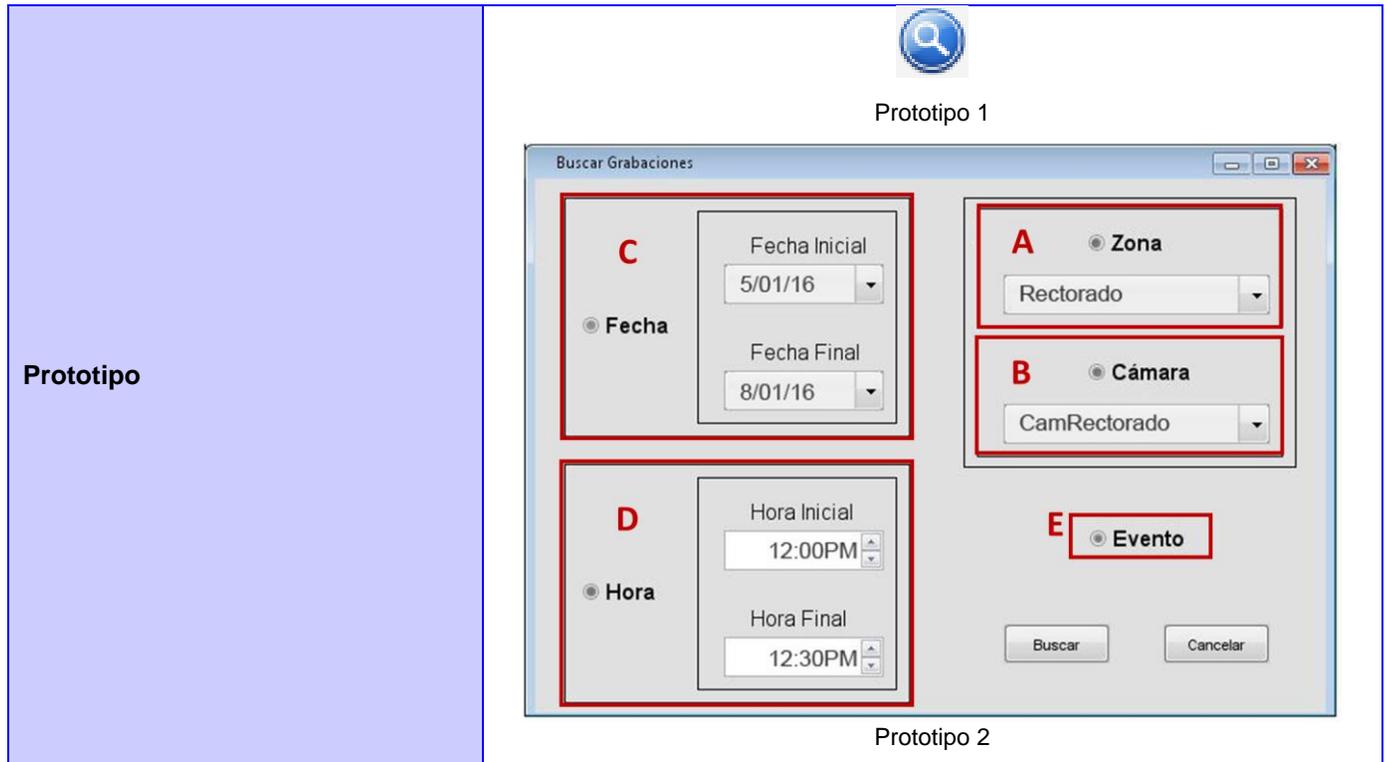


Tabla 5. Descripción del CU: Administrar conexión con el Gestor

<b>Objetivo</b>	Garantizar la comunicación del módulo recuperador con el Gestor.	
<b>Actores</b>	Usuario (inicia) Configurar y editar la conexión con el Gestor.	
<b>Resumen</b>	El CU inicia cuando el usuario configura o edita la conexión con el Gestor y finaliza cuando la configuración es editada o establecida.	
<b>Complejidad</b>	Media.	
<b>Prioridad</b>	Crítico.	
<b>Precondiciones</b>	Estado del Gestor "Disponible".	
<b>Postcondiciones</b>	Se establece la conexión del módulo con el Gestor	
<b>Referencia</b>	<b>RF19.</b> Configurar conexión con el gestor desde el recuperador. <b>RF20.</b> Editar configuración.	
<b>Flujo de eventos</b>		
<b>Flujo básico: administrar conexión con el Gestor satisfactoriamente.</b>		
	<b>Actor</b>	<b>Sistema</b>
1	Selecciona en "Opciones" la funcionalidad "Configurar"	
2		Muestra la interfaz de configuración del recuperador.

3	Introduce la dirección IP del gestor y el puerto para establecer la conexión con el Gestor.	
4	Selecciona la opción "Aceptar"	
5		Verifica que los datos sean correctos y establece la conexión con el gestor. Termina el CU.
<b>Flujos alternos</b>		
<b>Nº Evento 4. El usuario selecciona la opción "Cancelar".</b>		
	<b>Actor</b>	<b>Sistema</b>
4.1		Cierra la interfaz de la configuración sin realizar cambios en el módulo. Termina el CU.
<b>Nº Evento 5. Los datos del Gestor son incorrectos</b>		
	<b>Actor</b>	<b>Sistema</b>
5.1		Muestra un mensaje de que no se estableció la conexión con el Gestor y deshabilita las funcionalidades del módulo.
<b>Relaciones</b>	CU Incluidos	No procede
	CU Extendidos	No procede
<b>Prototipo</b>	 <p>Prototipo 3</p>	

Tabla 6. Descripción del CU: Realizar Operaciones con el Reproductor

<b>Objetivo</b>	Realizar operaciones con el reproductor.
<b>Actores</b>	Recuperador: (Inicia) Reproducción de las grabaciones así como operaciones básicas

	(retroceder o avanzar, reproducir, pausar, detener, disminuir o aumentar velocidad, control de volumen).	
<b>Resumen</b>	El CU inicia cuando el actor selecciona el video grabado con el que desea trabajar, luego que el video se carga puede realizar las operaciones que ofrece el reproductor. El CU termina cuando la acción es visualizada por el recuperador.	
<b>Complejidad</b>	Media.	
<b>Prioridad</b>	Crítico.	
<b>Precondiciones</b>	Módulo Recuperador conectado al Gestor. Usuario autenticado en el sistema. El video grabado debe estar en el espacio de trabajo.	
<b>Postcondiciones</b>	Video en reproducción lenta o rápida. Video en reproducción con mayor o menor volumen de sonido. Retroceso o adelanto del video. Video en reproducción. Reproducción del video detenida.	
<b>Referencia</b>	<b>RF7.</b> Reproducir video grabado. <b>RF8.</b> Pausar video grabado. <b>RF9.</b> Detener video grabado. <b>RF10.</b> Control de volumen de reproducción. <b>RF11.</b> Control de velocidad de reproducción. <b>RF12.</b> Control de la reproducción.	
<b>Flujo de eventos</b>		
<b>Flujo básico: Realizar operaciones con el reproductor</b>		
	<b>Actor</b>	<b>Sistema</b>
1	Selecciona el video grabado que desea reproducir o lo arrastra hacia el espacio de trabajo.	
2	Selecciona una de las siguientes opciones: <ul style="list-style-type: none"> <li>▪ Control de velocidad de reproducción (ver Sección 1: Control de velocidad de reproducción). (A)</li> <li>▪ Control de la reproducción (ver Sección 2: Control de la reproducción). (B)</li> <li>▪ Controlar el volumen de la grabación (ver Sección 3: Controlar el volumen de la</li> </ul>	

	grabación). (C)	
	<ul style="list-style-type: none"> <li>▪ Reproducir (ver Sección 4: Reproducir) (D)</li> <li>▪ Pausar (ver Sección 5: Pausar) (Ver Prototipo 4)</li> <li>▪ Detener (ver Sección 6: Detener) (E)</li> </ul>	
3		<p>Reproduce el video cumpliendo con las opciones seleccionadas por el usuario.</p> <p>Termina el CU.</p>
<b>Sección 1: Control de velocidad de reproducción.</b>		
<b>Flujo básico: Control de velocidad de reproducción.</b>		
	<b>Actor</b>	<b>Sistema</b>
2		Dirige la velocidad de reproducción, acelerando o retrasando, según seleccione el usuario.
<b>Sección 2: Control de la reproducción.</b>		
<b>Flujo básico: Control de la reproducción.</b>		
	<b>Actor</b>	<b>Sistema</b>
2		Dirige la reproducción, adelantando o atrasando, según seleccione el usuario.
<b>Sección 3: Controlar el volumen de la grabación.</b>		
<b>Flujo básico: Controlar el volumen de la grabación.</b>		
	<b>Actor</b>	<b>Sistema</b>
2		Dirige la reproducción aumentando o disminuyendo el volumen.
<b>Sección 4: Reproducir.</b>		
<b>Flujo básico: Reproducir.</b>		
	<b>Actor</b>	<b>Sistema</b>
2		Cambia el icono del botón por la imagen de pausa, y reproduce la grabación.
<b>Sección 5: Pausar.</b>		
<b>Flujo básico: Pausar.</b>		
	<b>Actor</b>	<b>Sistema</b>
2		Cambia el icono del botón por la imagen de reproducir, y pausa la grabación.
<b>Sección 6: Detener.</b>		
<b>Flujo básico: Detener.</b>		

Actor		Sistema
2		Detiene la reproducción, comenzándola desde el principio.
Relaciones	CU Incluidos	No procede
	CU Extendidos	No procede
Prototipo	 <p>Prototipo 4</p> <p>Prototipo 5</p>	

## 2.4 Descripción de la propuesta de solución.

Una vez que el usuario se autentique se le muestra la interfaz principal del recuperador con un árbol de cámaras con los videos grabados del día anterior y las funcionalidades de “opciones” y “buscar”. La primera muestra las opciones de configurar conexión, mostrar trazas y cerrar sesión posibilitándole al usuario realizar cambios para la conexión con el Gestor, visualizar un registro con los principales eventos realizados en determinado día y cerrar sesión respectivamente. La funcionalidad de “buscar” permite al usuario realizar búsquedas según determinados criterios, ya sea por rango de fecha u hora, por cámara y por zona.

Si el usuario desea reproducir uno o varios videos grabados del día anterior, primeramente, debe establecer la vista del espacio de trabajo según la cantidad de videos que desea visualizar, luego coloca los videos seleccionados o el video seleccionado en el espacio de trabajo y lo reproduce. Dado el caso que sea un video grabado de otro día, selecciona la opción de “buscar” donde escoge el criterio que desee, el sistema accederá al servidor de almacenamiento audiovisual para obtener los videos grabados que correspondan con la búsqueda y luego los muestra al usuario actualizando el árbol de cámaras inicial.

## **2.5 Arquitectura del software**

La arquitectura del software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución. El objetivo principal de la arquitectura del software es aportar elementos que ayuden a la toma de decisiones y, al mismo tiempo, proporcionar conceptos y un lenguaje común que permitan la comunicación entre los miembros de un proyecto. Para conseguirlo, la arquitectura del software construye abstracciones, materializándolas en forma de diagramas comentados. (Casanova, 2004)

### **2.5.1 Arquitectura en capas**

El patrón arquitectónico en capas pertenece a la familia del estilo en llamada y retorno, donde cada capa proporciona servicios a la capa superior y se sirve de las prestaciones que le brinda la inferior. Al dividir un sistema en N-capas, cada capa puede tratarse de forma independiente, sin tener que conocer los detalles de las demás. La división de un sistema en capas facilita el diseño modular, la comunicación entre capas está basada en una abstracción que proporciona un bajo acoplamiento entre ellas y con alta cohesión internamente, lo cual posibilita variar de una forma sencilla diferentes implementaciones o combinaciones de capas. (de la Torre Llorente, y otros, 2010)

#### **Arquitectura 3 capas:**

Para el desarrollo del Módulo Recuperador para Xilema-Suria 2.0 que se propone se definió una arquitectura en tres capas como una variante del patrón N-capas. Las mismas son denominadas como capa de presentación, capa de negocio y capa de servicio. La capa de presentación es la que interactúa con el usuario, también se le denomina "capa de usuario", comunica y captura la información. Por otra parte, la capa de negocio es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. En esta investigación esta capa es la que se encarga de los procesos de recuperación y reproducción de los videos grabados. La capa de servicio es la encargada de

la comunicación con el gestor y el servidor de almacenamiento audiovisual, además de devolver datos a la capa de negocio.



Figura 8. Arquitectura del Módulo Recuperador para Xilema-Suria 2.0.

## 2.6 Diagrama de Clases del Diseño

El diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema. Esta muestra sus clases, atributos y las relaciones entre ellos. Son utilizados durante el proceso de diseño. A continuación se representan las clases agrupadas por paquetes que componen la solución:



## 2.7 Patrones de Diseño

Un patrón de diseño constituye una solución estándar para un problema común de programación en el desarrollo del software. Además, es una técnica muy eficaz para flexibilizar el código haciéndolo satisfacer ciertos criterios, así como permite una manera más práctica de describir ciertos aspectos de la organización de un programa. Entre los patrones de diseño más utilizados se encuentran los patrones GRASP (del inglés *General Responsibility Assignment Software Patterns*, Principios Generales para Asignar Responsabilidades) y los patrones GoF (del inglés *Gang of Four*, Pandilla de los Cuatro).

### 2.7.1 Patrones GRASP empleados

Los patrones de asignación de responsabilidades describen los principios fundamentales del diseño de objetos para la asignación de responsabilidades (Gamma, y otros, 2004). Teniendo en cuenta el patrón arquitectónico seleccionado, los patrones de diseño que se aplican durante el desarrollo del módulo recuperador son los siguientes:

**Experto:** es usado para la asignación de responsabilidades en el desarrollo de sistemas. El mismo puede utilizarse en el diseño de aplicaciones orientado a objetos. (Larman, 2000) En el módulo se evidencia este patrón en la clase *MainWindow*, ya que garantiza que cada clase cumpla con sus responsabilidades, según la información que contenga y las funcionalidades que se implementan.

**Creador:** guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es crear una instancia de una clase *A* hacia una *B* (Larman, 2000). En el módulo se evidencia en la clase *VidTreeWidget*, ya que es la encargada de crear el árbol de cámaras, lo cual se realiza haciendo objetos de varias clases como por ejemplo la clase *zona*.

**Alta Cohesión:** caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Significa que una clase tiene responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas. En el módulo se evidencian clases como *Register*, que es la encargada de mostrar las trazas en el sistema y colabora con la clase *Log*, encargada de registrar las trazas.

**Bajo acoplamiento:** el acoplamiento es una medida de la fuerza con que una clase está conectada a otras. Una clase con bajo acoplamiento no depende de muchas otras. Se utilizó para asignar responsabilidades a

otras clases reduciendo las dependencias al mínimo, ejemplo de ello en el módulo es la clase *Log* que tiene como principal objetivo almacenar las trazas.

**Controlador:** Permite asignar la responsabilidad de controlar el flujo de eventos a clases específicas del sistema. Este patrón facilita la centralización de actividades, no porque las realice sino porque las delega en otras clases con las que mantiene un modelo de alta cohesión. En el módulo se evidencia en la clase *VideoView* ya que es la encargada de pasarle la instrucción a *VidTreeWidget* de crear el árbol de cámaras y a la clase *SearchDesigner* a realizar las búsquedas de videos grabados.

### 2.7.2 Patrones GoF empleados

Los patrones GoF son agrupados en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento. Los de creación abstraen el proceso de creación de instancias, los estructurales se ocupan de cómo clases y objetos son utilizados para componer estructuras de mayor tamaño y los de comportamiento atañen a los algoritmos y a la asignación de responsabilidades entre objetos. (Gamma, y otros, 2004) En el desarrollo del Módulo Recuperador para Xilema-Suria 2.0 se hace uso de los patrones siguientes:

**Observador (*Observer*):** 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 la clase *SearchDesigner* ya que cuando se realiza una búsqueda y se actualizan las zonas, actualiza además las cámaras asignadas a ella.

**Instancia única (*Singleton*):** como patrón de creación, está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. Este patrón ha sido escogido para ponerlo en uso en la conexión con las clases del *gestorclientproxy*, así como en la configuración del módulo recuperador, logrando tener una instancia global de estas.

## 2.8 Conclusiones del capítulo

En este capítulo se manifestaron las principales características del Módulo Recuperador para Xilema-Suria 2.0, cumpliendo las tareas de la investigación 4 y 5, lo que se evidencia en:

- ✓ La especificación de los requisitos funcionales y no funcionales, lo cual ha contribuido a una mejor organización del sistema, permitiéndole al desarrollador identificar sus funcionalidades básicas.
- ✓ La selección de la arquitectura del módulo, empleándose el patrón arquitectónico en tres capas como una variante de la arquitectura N-capas, facilitando la reutilización de las capas, la estandarización y la

contención de cambios, garantizando un mejor desempeño, portabilidad, flexibilidad y escalabilidad a la aplicación.

- ✓ La elaboración del diagrama de clases de diseño, obteniendo una visión estática del sistema, el cual sirvió de guía para la implementación del módulo.
- ✓ Se identificaron los patrones de diseño GRASP y GoF utilizados en el desarrollo de la aplicación, proporcionando la asignación de responsabilidades logrando un diseño de software que sirva de apoyo a la implementación del módulo.

## **Capítulo 3: “Implementación y prueba del Módulo Recuperador para Xilema-Suria 2.0”**

En el siguiente capítulo se describen los elementos relacionados con la implementación y los resultados de las pruebas realizadas al módulo. Se muestran los diagramas de componentes de implementación y de despliegue resultantes de esta etapa de construcción. Además, se describen las pruebas a las que fue sometido el módulo, 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. 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, a través del diagrama de despliegue, la estructura del sistema en ejecución y a través de los diagramas de componentes, las dependencias entre las partes de código del sistema.

#### **3.1.1 Diagrama de componentes de implementación**

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Los componentes representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes y bibliotecas cargadas dinámicamente. (Jacobson, y otros, 2000) En la siguiente figura se muestra el diagrama de componentes de implementación correspondiente a la aplicación:

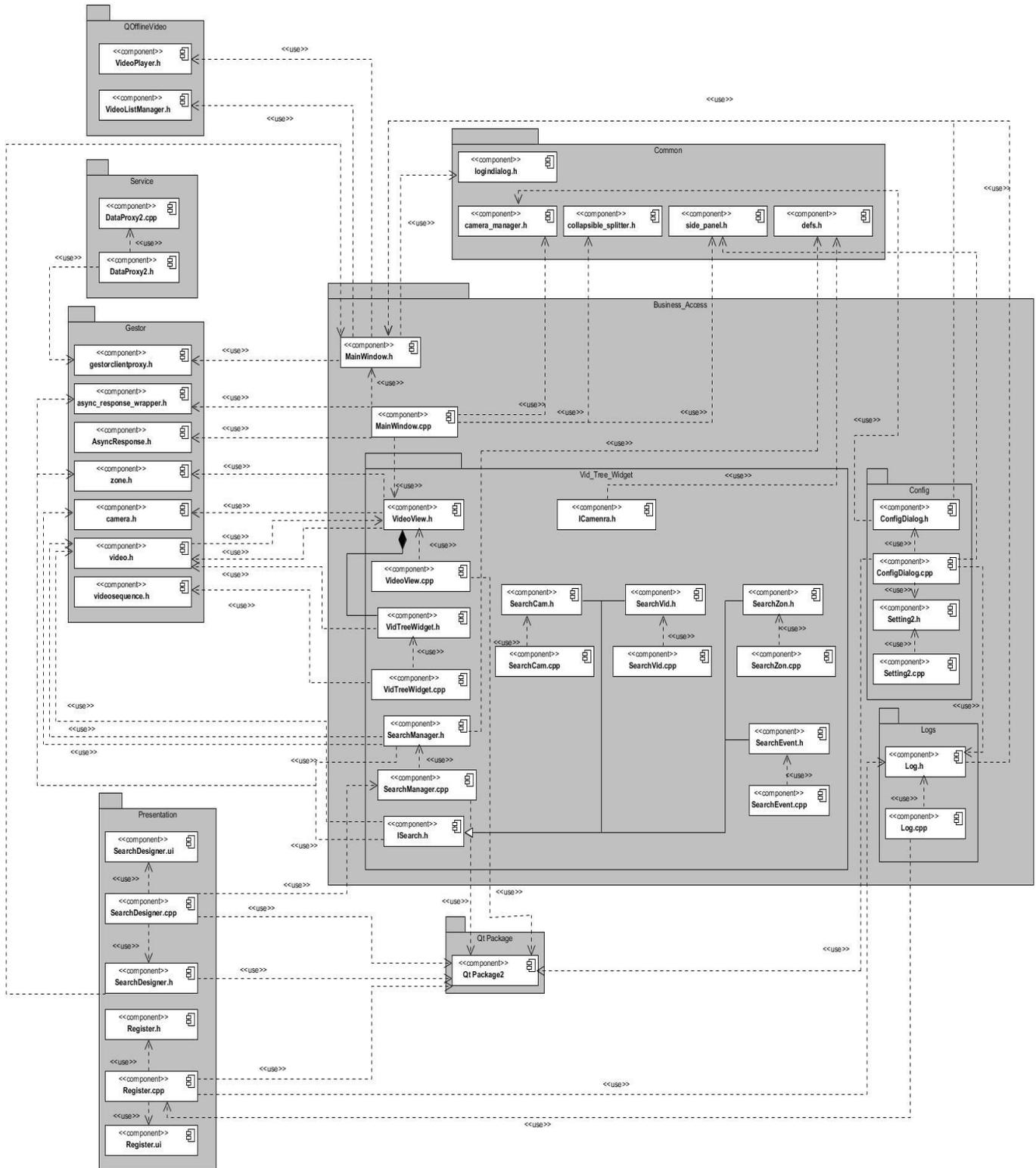


Figura 10. Diagrama de componentes del Módulo Recuperador para Xilema-Suria 2.0

### 3.1.2 Diagrama de despliegue

La vista de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los vínculos de comunicación entre ellos y las instancias de los componentes. El diagrama está compuesto por nodos, dispositivos y conectores. El propósito del modelo de despliegue es capturar la configuración de los elementos de procesamiento y las conexiones entre estos elementos en el sistema. (Pressman, 2010) El diagrama de despliegue del Módulo Recuperador para Xilema-Suria 2.0 consta de 4 elementos fundamentales:

**PC cliente:** Nodo que representa una computadora compuesta por el Módulo Recuperador para Xilema-Suria 2.0 y el cual se comunica con el módulo Gestor y el servidor de almacenamiento audiovisual a través del protocolo de red TCP/IP.

**Servidor compuesto por el módulo Gestor:** Nodo principal que contiene el módulo gestor, permite la comunicación entre el modulo recuperador y la base de datos a través del protocolo de red TCP/IP.

**Servidor de almacenamiento audiovisual:** Nodo que representa el servidor donde se almacenan las grabaciones.

**Servidor de base de datos:** Nodo que hace referencia al lugar donde se almacenan los datos del sistema de manera física.

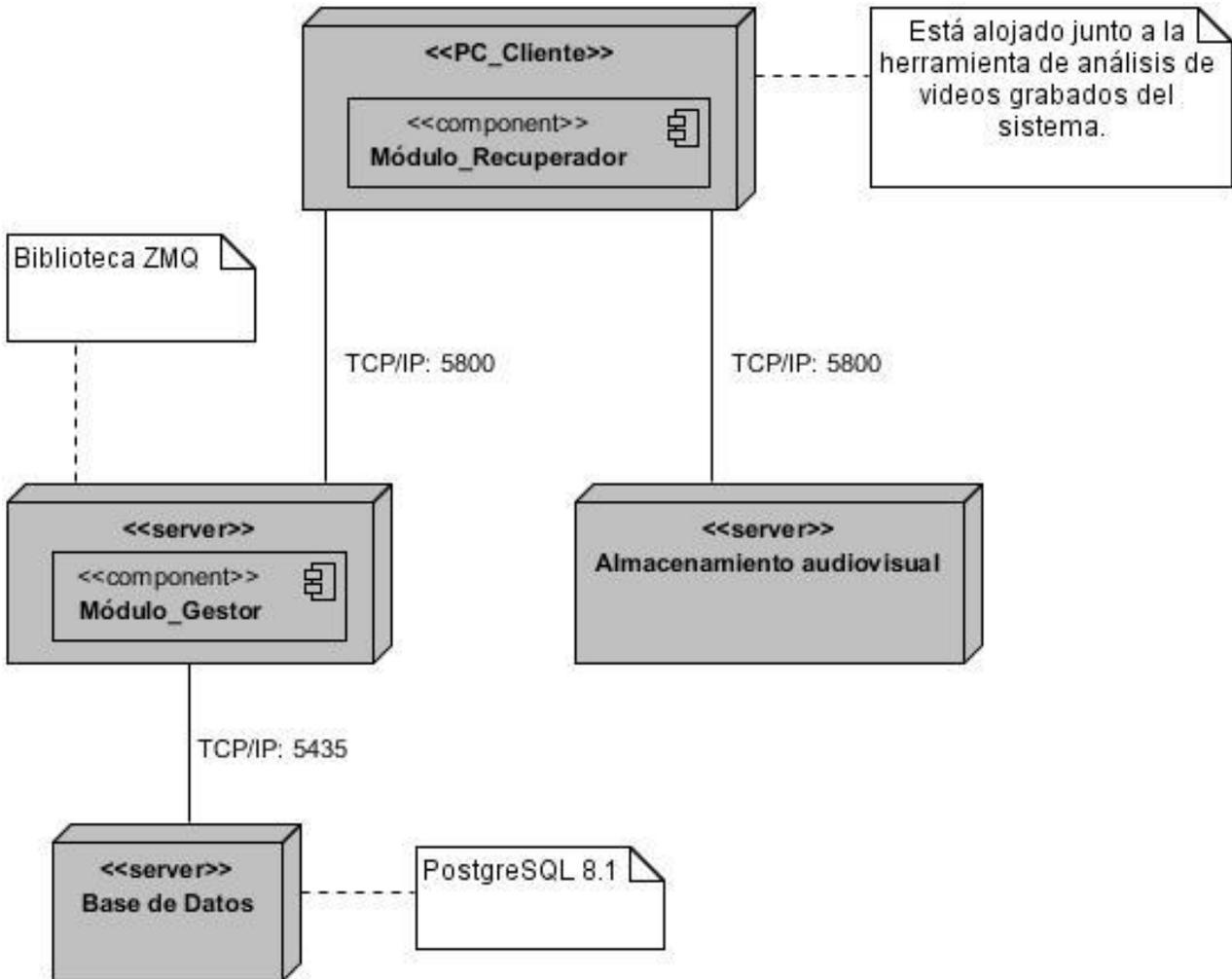


Figura 11. Diagrama de Despliegue del Módulo Recuperador para Xilema-Suria 2.0

## 3.2 Código fuente

El código fuente de un software es un conjunto de instrucciones que debe seguir la computadora para ejecutar dicho programa. Estas instrucciones son escritas en un lenguaje de programación que consiste en un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

### 3.2.1 Estándares de codificación

Los estándares de codificación no están enfocados a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Durante la

implementación del módulo, se utilizaron algunos estilos de codificación en busca de un estándar que aportara una mayor organización al código.

#### Principios generales y estándares utilizados:

- Los nombres de cada uno de los elementos del programa deben ser significativos; su nombre debe explicar en lo posible el uso del elemento.
- No manejar en los programas más de una instrucción por línea.

#### Ejemplo:

```
if(ui->chCamera->checkState())
{
    ui->cBCamera->setEnabled(true);
    ui->cBCamera->clear();
    if(ui->chZone->checkState())
    {
        foreach (Zone::Ptr zone, zones) {
            if(zone->name() == ui->cbzone->currentText())
            {
                zoneIDs=ZonHijos(zone->name());
            }
        }
    }
}
```

- Para los nombres de las clases e interfaces se usó el método Pascal con la primera letra en mayúscula, para las variables la misma notación en Pascal pero con la primera inicial en minúscula

#### Ejemplo:

Clase: MainWindow, SearchDesigner

Métodos: MainWindow::**configure** (), void SearchDesigner::**setCameraCollection** ()

### 3.3 Pruebas

Una de las partes más importantes en el desarrollo del software son precisamente las pruebas. Las cuales consisten en un proceso de ejecución de un programa con la intención de descubrir los fallos que puedan existir en la implementación, debe mostrar un conjunto de características para encontrar la mayor cantidad de errores con un mínimo de esfuerzo (Pressman, 2010). En esta etapa se realizan pruebas de Integración y de Caja Negra.

#### 3.3.1 Prueba de Caja Negra: Partición equivalente

Al Módulo Recuperador para Xilema-Suria 2.0 se le realizaron pruebas de caja negra pues estas se centran en encontrar casos en los que el módulo no funcione de manera correcta. Las pruebas de caja negra están especialmente indicadas en aquellos módulos que van a presentar una interfaz para interactuar con el usuario.

Partición equivalente es una técnica de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos a partir de los cuales pueden derivarse casos de prueba. La partición equivalente se esfuerza por definir un caso de prueba que descubra errores, reduciendo así el número total de casos de prueba que deben desarrollarse. (Pressman, 2010)

Para probar el módulo desarrollado se utilizó la técnica de Partición de Equivalencia, ya que permite examinar los valores válidos e inválidos de las entradas existentes en el software. Para la realización de dicha técnica se requirió el diseño de los casos de prueba en correspondencia con las descripciones de los casos de uso. A continuación se presentan los casos de prueba utilizados para los casos de uso más importantes del módulo:

### **Descripción de los Casos de prueba:**

#### **CU3: Realizar Búsquedas.**

**Descripción general:** El CU inicia cuando se solicita realizar búsquedas de grabaciones y termina cuando se le muestra el listado de grabaciones.

**Condiciones de ejecución:** Módulo Recuperador conectado al Gestor y usuario autenticado en el sistema.

**Sección 1:** Realizar Búsquedas.

Tabla 7. Descripción del caso de prueba: CU Realizar búsqueda (Sección 1)

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1: Buscar grabaciones correctamente.	El operador selecciona la opción de buscar grabaciones, el sistema muestra una ventana con criterios de búsqueda donde el usuario elige cuáles o cuál desea para recuperar grabaciones y oprime el botón Buscar, seguidamente el sistema actualiza el árbol de cámaras mostrando las grabaciones correspondientes a la búsqueda realizada.	El sistema muestra las grabaciones correspondientes a la búsqueda realizada por el recuperador, actualizando el árbol de cámaras.	Módulo Recuperador/ clic en el botón "Buscar grabaciones"/ clic en el botón "Buscar".
EC 1.2: Cerrar Buscar grabaciones	El operador selecciona el botón "Cancelar". El sistema cierra la ventana "Buscar" y mantiene abierto el Recuperador con los controles de reproducción inhabilitados y el árbol de cámaras sin actualizarse.	El sistema Cierra la ventana de Buscar grabaciones y mantiene abierto el Recuperador con los controles de reproducción inhabilitados y sin actualizar el árbol de cámaras.	Módulo Recuperador/ clic en el botón "Buscar grabaciones"/ clic en el botón "Cancelar".

**Sección 2:** Realizar búsqueda por rango de fecha y hora.

Tabla 8. Descripción del caso de prueba: CU Realizar búsqueda (Sección 2)

Escenario	Descripción	Respuesta del sistema	Flujo central
<p>EC 2.1: Buscar grabaciones definiendo rango de fecha y hora correctamente.</p>	<p>El operador selecciona la opción de buscar grabaciones, el sistema muestra una ventana con criterios de búsqueda donde el usuario elige un rango de fecha y hora, luego oprime el botón Buscar. Seguidamente el sistema actualiza el árbol de cámaras mostrando las grabaciones correspondientes al rango de fecha y hora seleccionado.</p>	<p>El sistema muestra las grabaciones correspondientes a la búsqueda realizada por el recuperador, actualizando el árbol de cámaras.</p>	<p>Módulo Recuperador/ clic en el botón "Buscar grabaciones"/ clic en el botón "Fecha"/ selección de fecha inicial/ selección de fecha final/ clic en el botón "Hora"/ selección de hora inicial/ selección de hora final /clic en el botón "Buscar".</p>
<p>EC 2.2: Buscar grabaciones definiendo rango de fecha y hora correctamente.</p>	<p>El operador selecciona la opción de buscar grabaciones, el sistema muestra una ventana con criterios de búsqueda donde el usuario elige un rango de fecha y hora seleccionando uno de los dos rangos incorrectamente, luego oprime el botón Buscar. Seguidamente el sistema muestra un mensaje de error y vuelve a mostrar la ventana con los criterios para que el recuperador rectifique</p>	<p>El sistema muestra un mensaje de error y vuelve al estado anterior para que se rectifique el error cometido.</p>	<p>Módulo Recuperador/ clic en el botón "Buscar grabaciones"/ clic en el botón "Fecha"/ selección de fecha inicial/ selección de fecha final/ clic en el botón "Hora"/ selección de hora inicial/ selección de hora final /clic en el botón "Buscar".</p>

**CU4: Realizar operaciones con el reproductor.**

**Descripción general:** El CU inicia cuando el actor selecciona el video grabado con el que desea trabajar, luego que el video se carga puede realizar las operaciones que ofrece el reproductor. El CU termina cuando la acción es visualizada por el recuperador.

**Condiciones de ejecución:** Módulo Recuperador conectado al Gestor, usuario autenticado en el sistema y el video grabado debe estar en el espacio de trabajo.

**Sección 1:** Realizar operaciones con el reproductor.

Tabla 9. Descripción del caso de prueba: CU Realizar operaciones con el reproductor

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1: Visualizar grabaciones correctamente.	El recuperador selecciona la grabación y la arrastra hasta el área de trabajo, el sistema carga el video en el visor correspondiente y listo para reproducirlo.	El sistema muestra el área de trabajo con las grabaciones seleccionadas detenidas.	Módulo Recuperador/ árbol de cámaras/doble clic en la grabación o Módulo Recuperador/ árbol de cámaras/ clic sostenido a la grabación hasta el área de trabajo.
EC 1.2: Retroceder grabación.	El operador selecciona el botón de retroceder. El sistema retrocede la grabación.	El sistema retrocede la grabación.	Módulo Recuperador/ clic en el botón de retroceder.

EC1.3: Reproducir/Pausar	El operador selecciona el botón "Reproducir/Pausar". El sistema reproduce la grabación o pausa la grabación.	El sistema reproduce la grabación o pausa la grabación.	Módulo Recuperador/ clic en el botón de reproducir/pausar.
EC1.4: Detener.	El operador selecciona el botón "Detener". El sistema detiene la grabación.	El sistema detiene la grabación.	Módulo Recuperador/ clic en el botón de detener.
EC1.5: Avanzar.	El operador selecciona el botón de avanzar. El sistema adelanta la grabación.	El sistema adelanta la grabación.	Módulo Recuperador/ clic en el botón de avanzar.
EC1.6: Disminuir velocidad de reproducción	El operador selecciona el botón "Disminuir velocidad de reproducción". El sistema disminuye la velocidad de reproducción.	El sistema disminuye la velocidad de reproducción.	Módulo Recuperador/ clic en el botón de disminuir velocidad de reproducción.
EC1.7: Aumentar velocidad de reproducción	El operador selecciona el botón "Aumentar velocidad de reproducción". El sistema aumenta la velocidad de reproducción.	El sistema aumenta la velocidad de reproducción.	Módulo Recuperador/ clic en el botón de aumentar velocidad de reproducción.
EC1.8: Visualizar grabaciones incorrectamente.	El operador desea visualizar una grabación y esta no se reproduce. El sistema envía un mensaje de error "No existe conexión".	El sistema envía un mensaje de error "No existe conexión".	

**CU10: Administrar conexión con el gestor.**

**Descripción general** El CU inicia cuando el usuario configura o edita la conexión con el Gestor y finaliza cuando la configuración es editada o establecida.

**Condiciones de ejecución:** Estado del Gestor “Disponible”.

**Sección 1:** Establecer la conexión con el Gestor.

Tabla 10. Descripción del caso de prueba: Administrar Conexión con el Gestor

Escenario	Descripción	Respuesta del sistema	Flujo central
EC1.1: Configurar recuperador correctamente.	El operador selecciona la opción Configurar en la ficha Opciones del Recuperador, el sistema muestra un formulario a llenar para la correcta conexión con el gestor. El operador entra los datos correctamente.	El sistema realiza la conexión con el gestor.	Módulo Recuperador/ clic en la pestaña Opciones/ Configuración/ clic en el botón “Aceptar”.
EC 1.2: Configurar recuperador incorrectamente	El operador selecciona la opción Configurar en la ficha Opciones del Recuperador, el sistema muestra un formulario a llenar para la correcta conexión con el gestor. El operador entra uno de los datos incorrectamente. El sistema muestra un signo de exclamación de color	El sistema no se conecta con el gestor por lo que el operador no podrá trabajar con la aplicación de recuperar.	Módulo Recuperador/ clic en la pestaña Opciones/ Configuración/ clic en el botón “Aceptar”.

	rojo en el campo con el dato incorrecto y deshabilita el botón "Aceptar".		
EC 1.3 Cancelar configuración.	El operador selecciona el botón "Cancelar". El sistema cierra la ventana "Configuración".	El sistema cierra la ventana "Configuración".	Módulo Recuperador/ clic en la pestaña Opciones/ Configuración/ clic en el botón "Cancelar".

**Resultado de las pruebas:**

Durante el transcurso de la ejecución de las pruebas de caja negra se identificaron siete no conformidades (NC) en 3 iteraciones

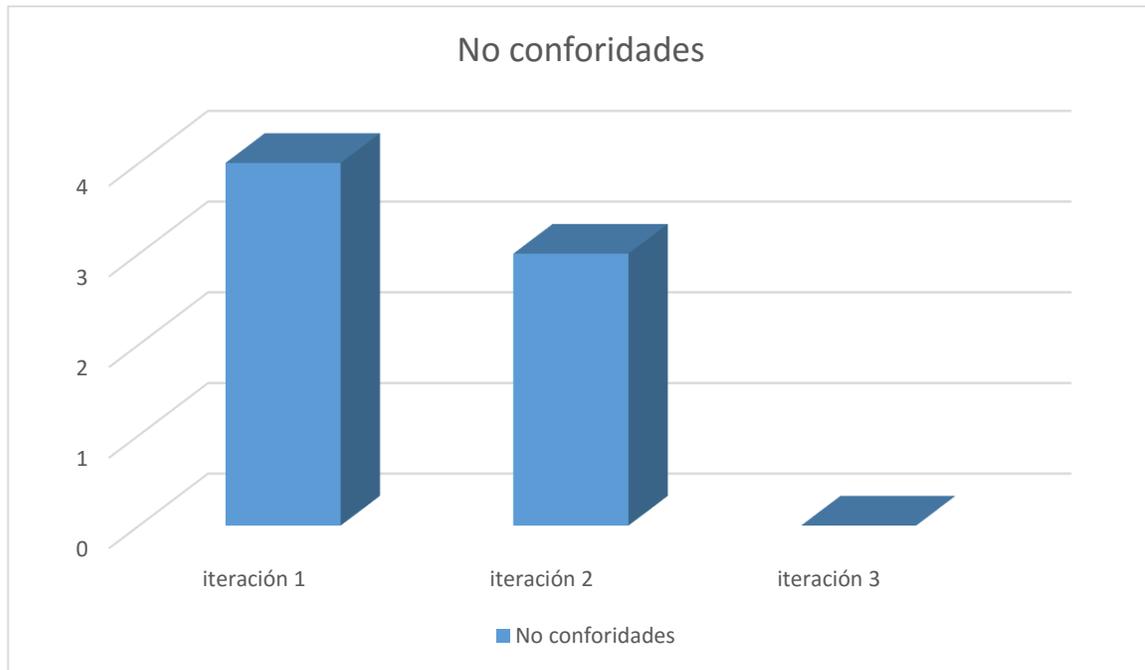


Figura 12. Gráfico de NC de las pruebas de caja negra

En la primera iteración se encontraron errores de validación y errores visuales relacionados con el diseño de las interfaces, obteniéndose un total de 4 NC. Luego de corregir los errores de la primera iteración se le da paso a una segunda iteración, donde se encontraron errores funcionales ya que el resultado mostrado al realizar determinada acción no fue el esperado. Finalmente, en la tercera iteración no se detectaron no conformidades, garantizando así el correcto funcionamiento y comportamiento de todos los casos de uso del módulo.

**3.3.2 Prueba de Integración**

La prueba de integración es una técnica sistemática para construir la arquitectura del software mientras, al mismo tiempo, se aplican las pruebas para descubrir errores asociados con la interfaz. Es necesario probar el sistema 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. (Pressman, 2010)

A menudo se tiende a intentar una integración que no sea incremental; es decir, se combinan todos los componentes por anticipado, se prueba el programa como un todo. El resultado puede ser un poco caótico debido a que se encuentra una gran cantidad de errores y la corrección es difícil porque resulta complicado identificar qué los provocó. Otra variante es aplicar la integración incremental, en la que el programa se construye y prueba en pequeños incrementos, en los cuales resulta más fácil detectar los fallos.

Para lograr una correcta integración del módulo recuperador con la herramienta de análisis de grabaciones se aplicó el enfoque incremental descendente. Se integran los componentes moviéndose hacia abajo por la jerarquía de control. La herramienta de análisis se incorpora en la estructura del recuperador; este llama a funcionalidades implementadas en la herramienta incluida y se valida que funcionen de manera correcta. En una primera iteración de pruebas de integración existieron cinco NC referentes a la integración, las mismas fueron resueltas y se les efectuó una regresión a las pruebas de integración. En la misma se identificó una NC, la cual fue solucionada. Luego se le aplicó una segunda iteración de pruebas en la que no se identificaron NC, quedando integrado el sistema de manera exitosa.

Además, se aplicó el enfoque incremental de integración ascendente ya que a medida que fueron desarrolladas las funcionalidades se agruparon y se coordinaron las entradas y salidas de los casos de prueba. Posteriormente se realizaron las pruebas al grupo de funcionalidades y al quedar probadas se continuó con el desarrollo de otras funcionalidades repitiendo el mismo enfoque. Una vez que se concluyó la implementación y se comprobó el correcto funcionamiento del módulo, se realizaron las pruebas para constatar su integración con el Gestor las que dieron como resultado que:

Para acceder al módulo inicialmente el usuario debe autenticarse, para garantizar que dicha funcionalidad se realice correctamente el sistema verifica los datos registrados a través del gestor. Inicialmente hubo errores debido a que el sistema no se conectaba con el Gestor. Al concluir la corrección del error que dio fin a la primera iteración se prosiguió a comenzar una nueva iteración para comprobar si existían nuevos errores. Nuevamente se efectúa la autenticación, logrando entrar exitosamente al sistema. Seguidamente se comprobó la funcionalidad de realizar búsqueda que inicialmente no mostraba los resultados de la misma, ya para la siguiente iteración se lograron los resultados esperados. Finalmente se comprobaron las restantes funcionalidades del módulo como la de gestionar las vistas, mostrar metadatos, cerrar sesión y configuración con el Gestor, sin ocurrir ningún error. De esta manera culmina exitosamente la prueba comprobándose la correcta integración entre el Gestor y el módulo recuperador.

### **3.4 Conclusiones del capítulo**

En este capítulo se definieron temas relacionados con los procesos de implementación y prueba del Módulo Recuperador para Xilema-Suria 2.0, cumpliendo las tareas de la investigación 6 y 7, lo que se evidencia en:

- ✓ La elaboración del diagrama de componentes, el cual permitió describir los elementos representados en el diseño en forma de componentes de software.
- ✓ La confección del diagrama de despliegue, sirviendo de guía para la instalación del Módulo Recuperador para Xilema-Suria 2.0.
- ✓ Las pruebas realizadas validaron el funcionamiento del sistema de acuerdo con los requisitos y su correcta integración con la herramienta de análisis y con el Módulo Gestor.

## **Conclusiones Generales**

Con el presente trabajo de diploma se logró dar cumplimiento a las tareas de investigación propuestas, logrando obtener como resultado un módulo recuperador para el sistema Xilema-Suria. La investigación permitió arribar a las siguientes conclusiones:

La caracterización y revisión del estado del arte de los sistemas de video vigilancia y la recuperación de grabaciones, 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 de solución.

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 módulo recuperador y el Sistema de Video Vigilancia.

La utilización del patrón arquitectónico N-Capas permitió dividir el componente en tres capas: Presentación, Negocio y Servicio, admitiendo 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 al módulo recuperador desarrollado demuestran el cumplimiento de los requisitos especificados, su correcto funcionamiento y que está apto para ser integrado al Sistema de Video Vigilancia Xilema-Suria.

Se logró desarrollar un módulo capaz de recuperar grabaciones y reproducirlas, además de realizar el registro de las trazas; facilitando el proceso de recuperación de las grabaciones realizadas en Xilema-Suria 2.0 y la visualización de los materiales audiovisuales.

## **Recomendaciones**

Terminada la investigación y con sus objetivos cumplidos, se han identificado un conjunto de elementos que serían de mucha utilidad para el sistema su desarrollo futuro, por lo que se recomienda:

- ✓ Integrar el sistema con varios servidores de almacenamientos de grabaciones.
- ✓ Integrar el sistema con un Sistema de Información Geográfica (SIG).

## Bibliografía

- Reinoso Medina, Leonid , Niubó Hernández, Ariel y Benavides Angulo, Francisco . 2016.** Plataforma de Video Vigilancia XYMA SAFE VISION. [En línea] 2016.
- Albers, Rob, Wijnhoven, Rob y Jaspers, Egbert. 2015.** *Video retrieval system, method and computer program for surveillance of moving objects.* 2015.
- Axis. 2014.** Axis. [En línea] 2014. [Citado el: 29 de noviembre de 2015.] <http://www.axis.com/ca/en/products/axis-camera-station/support-and-documentation>.
- C++. 2014.** C++. [En línea] 2014. [Citado el: 27 de enero de 2016.] <http://www.cplusplus.com/>.
- Casanova, Josep. 2004.** desarrolloweb.com. *Usabilidad y Arquitectura del Software.* [En línea] 2004. [Citado el: 17 de febrero de 2016.] <http://www.desarrolloweb.com/articulos/1622.php>.
- Case, Albert F. 2001.** *Computer-aided software engineering (CASE): technology for improving software.* New York : s.n., 2001.
- CEUNET, Bibliotecas Universitarias. 2007.** CEU Biblioteca. [En línea] 2007. [Citado el: 19 de diciembre de 2015.] <http://www.bibliotecaceu.es/servicios/mediateca>.
- Codebox. 2010.** Codebox. *FRAMEWORK.* [En línea] 2010. [Citado el: 27 de enero de 2016.] <http://www.codebox.es/glosario>.
- de la Torre Llorente, César, y otros. 2010.** *Guía de Arquitectura N-Capas orientada al Dominio con .NET 4.0.* 2010.
- definicion. 2015.** definicion.de. *definicion.de/plugin/.* [En línea] 2015. [Citado el: 15 de noviembre de 2015.]
- Dointech. 2016.** Automatización, Seguridad y Control, DOINTECH. [En línea] 2016. <http://www.dointech.com.co>.
- Fowler, Martin. 2002.** *Patterns of enterprise application architecture.* 2002.
- Fritzler, Luciano. 2011.** *Entorno de Desarrollo Integrado.* Buenos Aires : s.n., 2011.
- Gamma, Erich, y otros. 2004.** *Design Patterns.* 2004.
- García, Francisco Javier. 2010.** *Videovigilancia: CCTV usando videos IP.* s.l. : Vértice, 2010.
- Hernández León, Rolando Alfredo y Coello González, Sayda. 2002.** *"El Paradigma Cuantitativo de la Investigación Científica".* La Habana : Editorial Universitaria, 2002.

- Hintjens, Pieter. 2002.** the guide. [En línea] 2002. [Citado el: 28 de enero de 2016.] <http://zguide.zeromq.org/page:all>.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000.** *El proceso unificado de desarrollo de software*. Madrid : Pearson Educacion, 2000.
- Larman, Craig. 2005.** El Unified Modeling Language, UML. *UML y Patrones*. Mexico : Prentice Hall, 2005.
- . **2000.** *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Madrid : Prentice Hall, 2000.
- Microsoft Corporation. 2011.** Microsoft Corporation. [En línea] 2011. [Citado el: 27 de enero de 2016.] [www.microsoft.com](http://www.microsoft.com).
- Nilsson, Fredrik. 2011.** [http://www.infosectoday.com/Articles/Video\\_Surveillance\\_Systems.htm](http://www.infosectoday.com/Articles/Video_Surveillance_Systems.htm). [En línea] 2011. [Citado el: 28 de noviembre de 2015.]
- Pressman, Roger. 2010.** *Ingeniería de Software. Un enfoque Práctico*. 2010.
- Pressman, Roger S. 2010.** *Software engineering: a practitioner's approach*. 7th ed. 2010.
- QT Centre. 2014.** QT Centre. [En línea] 2014. [Citado el: 27 de enero de 2016.] <http://www.qtcentre.org>.
- RAE. 2014.** RAE. [En línea] 2014. [Citado el: 25 de noviembre de 2015.] <http://dle.rae.es/?id=bmtXm9x>.
- Rodríguez, Fernando García. 2016.** Tecnología de la Seguridad. [En línea] 2016. <http://serviciostc.com/>.
- Sermar, S.L. 2016.** SecurOS. Video Management Systems. [En línea] 2016. <http://www.isstechnology.eu/>.
- Sommerville, Ian. 2005.** *Ingeniería del Software Séptima Edición*. Madrid : Pearson Educación.SA, 2005.
- SyG Seguridad. 2012.** [sygseguridad.](http://www.sygseguridad.com.ar/nota_tecnica.asp?id=9) [En línea] 2012. [http://www.sygseguridad.com.ar/nota\\_tecnica.asp?id=9](http://www.sygseguridad.com.ar/nota_tecnica.asp?id=9).
- Tolosa, Gabriel H. y Bordignon, Fernando A. 2007.** *Introducción a la Recuperación de Información*. Argentina : s.n., 2007.
- UCI. 2015.** Metodología de desarrollo para la Actividad productiva de la UCI 1.2. [En línea] 2015.
- VMS. 2005.** <https://www.exacq.com>. [En línea] 2005. [Citado el: 4 de diciembre de 2015.]