



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
Facultad 6



Módulo Recuperador para el Sistema de Video Vigilancia

Trabajo de diploma para optar por el título de Ingeniero en
Ciencias Informáticas.

AUTOR: Oslanier Villa Castillo

TUTOR: Ing. Fernando Echemendia Tourt

Ciudad de la Habana / Año 53 de la Revolución

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los ____ días del mes de _____ del año ____2011____.

Autor:

Oslanier Villa Castillo

Tutor:

Ing. Fernando Echemendia Tourt.

DATOS DE CONTACTO

Tutor:

Ing. Fernando Echemendia Tourt (fechemendia@uci.cu)

Graduado de Ingeniero en Ciencias Informáticas en el año 2008. Se desempeña como desarrollador e investigador en el proyecto Video Vigilancia del departamento de Señales Digitales del Centro Geoinformática y Señales Digitales (GEySED). Imparte las asignaturas optativas Procesamiento de Imágenes Digitales I y II.

AGRADECIMIENTOS

Le agradezco a mi madre por estar siempre a mi lado y por dármele todo en la vida,
A mis abuelos Tania y Enrique que son mi gran alegría,
A Tata a Iris y Alain por ser mi inspiración,
A mi hermano por ser un hermano aunque nos criáramos separados,
A mi papa por ayudarme a entender como es la vida,
A Susy por estar siempre a mi lado y animarme incluso en los momentos en que comienzo a ser
insoportable,
A mis suegros por ayudarme en todo,
A mi abuelita fina y a Eladito,
A mi tutor por soportarme y tirarme unos cuantos salves,
A la gente del barrio el Belo, Alfro, Enier, Sanga, Robe, Majua,
A mi gente del 9108 que fueron los primeros en conocerme,
Mis mujeres del 107 Eli, Lusmey, Merlin la flaki, Lisandra, Maylen, Ochoa, Yara, Diana,
Grechin,
Mi gente de la P.M.M. Osmelito, Franio, Rodo, El negro (Diosbel),
A los pentiumBoys mi gente del 112 Pitbull, Fresa, Carlo, Rosabal, Fide, El Misu, Rene, Yasiel,
Trista,
, Paca y Dussu y a toda mi lista de contacto del jabber,
A Diana Naranjo que me ayudó mucho,
Y a María por estar siempre ahí a toda hora para hacernos la UCI más amena.

DEDICATORIA

Le dedico este trabajo:

A mi madre que es la mujer y la personita que más quiero yo en este mundo,
A mis abuelitos Tania y Enrique que han luchado mucho para que todos en la familia salgamos
adelante,

A Iris y Tata que aunque están lejos sé que no dejan de pensar en mí,

A primito Alain que es mi vida y mi inspiración,

A toda mi familia y a las personas que ahora también los son,

Por quererme y soportarme les regalo mi título de ingeniero.

RESUMEN

Debido al inmenso desarrollo tecnológico acontecido en las últimas décadas, los sistemas dedicados a la vigilancia por video, han incorporado numerosas funcionalidades y dispositivos de hardware con el objetivo de viabilizar el uso de los mismos, lo que a su vez ha provocado que estos sistemas tengan precios, cada vez más elevados, en el mercado internacional. Debido a las limitaciones económicas existentes en nuestro país, la inmensa mayoría de estas soluciones son prácticamente inaccesibles por lo que en la Facultad 6 de la Universidad de las Ciencias Informáticas se lleva a cabo la construcción de un sistema de video vigilancia llamado *Suria SIV* capaz de satisfacer las necesidades de las entidades nacionales.

Actualmente el sistema, al encontrarse en proceso de desarrollo, carece del módulo que permita la recuperación de los videos almacenados en la base de datos siendo el objetivo de la presente investigación la planificación de las actividades necesarias para la implementación y puesta en marcha de un *Módulo Recuperador* para *Suria SIV*. Este módulo, además de ser una parte fundamental para este tipo de sistemas, pretende disminuir el tiempo empleado en las actividades de búsqueda y visualización de los videos almacenados agilizando y facilitando la interacción con el sistema de video vigilancia.

En el presente documento se presenta la fundamentación teórica que sustenta la realización de este trabajo. Se valora el estado del arte de los sistemas video vigilancia así como el estudio de algunas de las tecnologías, herramientas y metodologías más utilizadas en la actualidad para el desarrollo de este tipo de aplicaciones. Además se analiza el negocio, se desarrolla un modelo del diseño y se obtiene la aplicación final.

PALABRAS CLAVES

Sistema de video vigilancia, módulo recuperador, seguridad.

ÍNDICE DE TABLAS

Tabla 1. Evolución de los sistemas de video vigilancia	5
Tabla 2. Actor del Sistema	24
Tabla 3. CUS_AutenticarUsuario.....	25
Tabla 4. CUS_ExportarVideo	26
Tabla 5. CUS_BuscarVideo	27
Tabla 6. CUS_VisualizarVideo	28
Tabla 7. SC 1: Autenticas Usuario	54
Tabla 8. SC 2: Exportar videos.....	56
Tabla 9. SC 3: Buscar videos.....	58
Tabla 10. Resultados de las pruebas aplicadas.....	59

ÍNDICE DE FIGURAS

Figura 1. Sistemas de CCTV analógico usando VCR	6
Figura 2. Sistemas de CCTV analógico usando DVR de red.....	7
Figura 3. Sistemas de video IP que utilizan servidores y cámaras IP	7
Figura 4. Reproductor de Vivotek ST7105	8
Figura 5: Reproductor del sistema D-Link.....	9
Figura 6. Sistema Axis Camera Station.....	10
Figura 7. Sistema para kit de cámaras Panasonic	10
Figura 8. Sistema ZoneMinder	11
Figura 9. Descripción de la composición de RUP.....	18
Figura 10. Modelo de Dominio	21
Figura 11. Diagrama de Casos de Uso.....	28
Figura 12. Arquitectura en Pizarra	30
Figura 13. Patron <i>Modelo-Vista-Controlador</i>	31
Figura 14. Arquitectura en el módulo Recuperador	32
Figura 15. CU <i>Autenticar usuario</i>	33
Figura 16. CU <i>Buscar videos</i>	33
Figura 17. CU <i>Exportar videos</i>	34
Figura 18. CU <i>Visualizar video</i>	34
Figura 19. Diagrama de clases del diseño <i>CU_AutenticarUsuario</i>	35
Figura 20. Diagrama de secuencia <i>CU_AutenticarUsuario</i>	36
Figura 21. Diagrama de clases del diseño <i>CU_BuscarVideo</i>	37
Figura 22. Diagrama de secuencia <i>CU_BuscarVideo</i>	38
Figura 23. Diagrama de clases del diseño <i>CU_ExportarVideo</i>	39
Figura 24. Diagrama de secuencia <i>CU_ExportarVideo</i>	40
Figura 25. Diagrama de clases del diseño <i>CU_VisualizarVideo</i>	41
Figura 26. Diagrama de secuencia <i>CU_VisualizarVideo</i>	42
Figura 27. Diagrama de Despliegue	47
Figura 28. Diagrama de Componentes	48

Figura 29.Enfoque de diseño de pruebas de caja blanca 51
Figura 30.Enfoque de diseño de pruebas de caja negra..... 52

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA	5
1.1. Introducción	5
1.2. Evolución de los sistemas de recuperación de videos digitales.....	5
1.3. Estado del arte de los sistemas de recuperación de videos digitales.....	9
1.4. Tecnologías, herramientas y metodologías a utilizar para el desarrollo de la solución.12	
1.4.1. Tecnologías	12
1.4.2. Herramientas	13
1.4.3. Metodologías	15
1.5. Conclusiones del capítulo	18
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA	20
2.1. Modelo de Dominio	20
2.1.1. Conceptos Fundamentales.....	20
2.1.2. Diagrama del Modelo de Dominio	21
2.2. Propuesta de Sistema	21
2.3. Requisitos Funcionales del Sistema	21
2.3.1. Requerimientos Funcionales.....	21
2.4. Requerimientos no Funcionales	22
2.4.1. Requerimientos de Usabilidad	22
2.4.2. Requerimientos de Fiabilidad	23
2.4.3. Requerimientos de Eficiencia.....	23
2.4.4. Requerimiento de Soporte.....	23
2.4.5. Requerimiento de Funcionamiento	24
2.5. Definición de Casos de Uso	24
2.5.1. Definición de los actores.....	24
2.5.2. Especificación de los Casos de Uso del Sistema(CUS).....	24
2.5.3. Diagrama de Casos de Uso	28
2.5.4. Conclusiones del capítulo.	28
CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA.....	30
3.1. Arquitectura	30

3.2.	Modelo de Análisis.....	33
3.2.1.	Diagrama de clases de análisis.....	33
3.3.	Modelo de Diseño.....	35
3.4.	Descripción de las clases.....	42
3.5.	Conclusiones del capítulo.	45
CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBA.....		46
4.1.	Modelo de Implementación.....	46
4.1.1.	Diagrama de Despliegue	46
4.1.2.	Diagrama de Componentes.	47
4.2.	Validación de la Solución Propuesta.....	48
4.2.1.	Pruebas de Software.....	49
4.2.2.	Técnicas de Prueba.	50
4.2.3.	Diseño de los casos de pruebas (DCP) Caja Negra. Partición equivalente.....	52
4.2.4.	Resultados de las pruebas aplicadas.....	58
4.3.	Conclusiones.	59
CONCLUSIONES GENERALES		60
RECOMENDACIONES		61
REFERENCIA BIBLIOGRÁFICA		63

INTRODUCCIÓN

La seguridad siempre ha sido un factor de suma importancia para el hombre. Con los grandes adelantos tecnológicos, el incremento del volumen de los datos y la llegada de las Tecnologías de la Informática y las Comunicaciones (TIC) hicieron su aparición los sistemas dedicados a la vigilancia exhaustiva de viviendas, negocios y edificios. Estos sistemas fueron concebidos para garantizar un control más efectivo de los métodos de protección planificados por cada entidad y para asistir al personal encargado de la vigilancia en situaciones específicas como la ubicación rápida de alguna actividad delictiva, intrusión externa o de algún problema determinado.

A lo largo de los años los sistemas de video vigilancia evolucionaron partiendo de los sistemas CCTV¹ analógicos surgidos hace alrededor de 25 años, donde varias cámaras eran conectadas por medio de cables coaxiales a monitores y a grabadores de cinta. Con la llegada de las cámaras para UTP² se hizo posible incorporarlas a la infraestructura de red de cualquier edificio pues estas cámaras podían ser instaladas en cualquier ubicación que existiera una toma de red. En este tipo de sistema se continuaron empleando los grabadores de cinta magnética. El cableado viajaba hasta un ³multiplexor con conectores RJ45⁴ y a las cámaras tradicionales con conectores coaxiales se les incorporaban baluns⁵ para convertir la señal de un cable coaxial (no balanceada) a la del cable par trenzado (balanceada). El advenimiento de la era digital trajo consigo los grabadores de video digital (DVR). Estos resolvieron muchos de los problemas que generaban las cintotecas⁶ de medios magnéticos dando lugar a un nuevo sistema de video vigilancia CCTV basado en IP, con cámaras y servidores IP que pueden colocarse en cualquier punto de la infraestructura de una red. En este punto, ya se contaba con tecnología suficiente para almacenar las secuencias de video de forma digital pero aún se hacía engorrosa la búsqueda de estos archivos pues esta actividad se hacía de forma manual. El operario del sistema, en caso de que ocurriera algún delito y debiera consultar una grabación, debía revisar cada uno de los videos grabados hasta encontrar la secuencia deseada. En la actualidad, con la creación de los sistemas de tercera generación se elimina el problema de la revisión manual

¹ Sistema de televisión de circuito cerrado

² Cable de par trenzado

³ Dispositivo que puede recibir varias entradas y transmitir las por un medio de transmisión compartido. Para ello divide el medio de transmisión en múltiples canales, para que varios nodos puedan comunicarse al mismo tiempo.

⁴ Interfaz física comúnmente usada para conectar redes de cableado estructurado.

⁵ Dispositivo adaptador que convierte líneas de transmisión simétricas en asimétricas.

⁶ Almacén de los medios magnéticos

de las pistas de video ya que se incorporan módulos como el de recuperación de archivos grabados que permite agilizar todo el proceso de búsqueda y visualización teniendo en cuenta diversos criterios como son la fecha de grabación y la cámara de la que fue tomada la secuencia de video.

En nuestro país se han llevado a cabo grandes inversiones para la compra de equipamiento con tecnología de punta para brindar una mejor atención a la población en centros comerciales, instalaciones hospitalarias, centros de enseñanza de todo tipo entre otros. Debido a los altos precios de este equipamiento aumenta el potencial delictivo alrededor de estas instituciones, lo que hace necesario un sistema encargado de controlar las zonas que llegan a ser tan extensas que el personal encargado de la seguridad no es capaz de cubrirlas en su totalidad.

Actualmente, 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. En nuestro país, a pesar de existir una demanda creciente de este tipo de soluciones, las empresas nacionales no pueden contar con los sistemas que producen este tipo de compañías ya que los precios para la adquisición de estos productos son sumamente elevados además de que la mayoría de estos sistemas son desarrollados para ser usados de manera conjunta con los kit de cámaras producidos por la misma compañía que los fabrica. Otro inconveniente que presentan estas soluciones es que en su mayoría no son multiplataforma y funcionan solo sobre software propietario aunque existen sistemas como el *ZoneMinder* que trabaja sobre sistemas *GNU/Linux* pero generalmente presentan dificultades en cuanto a usabilidad y accesibilidad.

Debido a esto, una de las mayores empresas de desarrollo de software del país, *DATYS* incursionó en el desarrollo de un sistema de video vigilancia basado en tecnología IP llamado *XYMA SAFE VISION*. Dicho sistema, a pesar de ser un gran avance en cuanto a producción de software nacional respecta, no llega a satisfacer las necesidades de empresas menores y pequeños comercios ya que este tiene un alto precio de adquisición y trabaja sobre software propietario obviando una de las premisas fundamentales que rigen el desarrollo tecnológico en la isla que es la migración paulatina a software libre. Por ello en la Facultad 6 de la Universidad de las Ciencias Informáticas se decidió desarrollar el sistema de video vigilancia *SuriaVision*

con el que se pretende llevar al mercado una solución multiplataforma desarrollada totalmente sobre software libre con un alto grado de modularidad, flexibilidad y adaptabilidad.

Suria cuenta con varios módulos y entre ellos se encuentra el módulo *SuriaRetriever* que es el encargado de realizar las tareas de búsqueda y visualización de los videos almacenados por el sistema y en el presente documento se detalla el proceso de desarrollo del mismo.

A partir del análisis de las situaciones anteriormente descritas se identificó el siguiente **problema a resolver**: La carencia de un mecanismo que permita buscar y visualizar los videos digitales almacenados por el Sistema de Video Vigilancia, provoca retrasos en las actividades realizadas sobre los mismos.

Para brindar una solución adecuada y eficiente que estuviese acorde con la problemática planteada se decidió centrar la investigación en el siguiente **objeto de estudio**: el proceso de recuperación de medias en los sistemas de video vigilancia. Para ello se identificó el siguiente **campo de acción**: la informatización de los procesos búsqueda y visualización de videos digitales en el Sistema de Video Vigilancia.

Se propone como **idea a defender** el desarrollo de una aplicación que permita buscar y visualizar los videos almacenados por el sistema de Video Vigilancia evitará el gasto innecesario de tiempo en las operaciones realizadas sobre los archivos grabados.

Con la realización de este trabajo se pretende alcanzar el **objetivo general** de desarrollar un sistema que permita realizar las tareas de búsqueda y visualización de los videos digitales obtenidos por el Sistema de Video Vigilancia.

Como **métodos científicos de investigación** se utilizaron **métodos teóricos** como el **analítico – sintético**, mediante el análisis se separó, en sus diversas partes y cualidades, los elementos que componen el Sistema de Video Vigilancia con el fin de lograr un mayor entendimiento del funcionamiento interno del mismo. La síntesis unifica los elementos previamente analizados y permite identificar las relaciones fundamentales y características generales entre ellos. Se empleó además el método de **análisis histórico - lógico** ya que permite el estudio de la evolución y desarrollo del método empleado para la búsqueda, recuperación y visualización de los videos grabados por los sistemas de video vigilancia y su condicionamiento a los diferentes períodos en el decursar de su historia, poniendo de

manifiesto la lógica de las leyes generales de su funcionamiento. La estructura lógica del Sistema de Video Vigilancia necesita ser modelada (para lo que se utiliza el **método modelación**) a partir de abstracciones (diagramas derivados del proceso de modelado del negocio) que se crean con el fin de reflejar la forma en la que se encuentran distribuidos los distintos módulos que conforman el sistema brindando una percepción clara y concreta de la realidad.

Se emplearon además **métodos empíricos** ya que estos posibilitan revelar las características fundamentales del objeto de estudio a través de procedimientos prácticos. De ellos se seleccionó la **observación científica** debido a que este método permite conocer la realidad mediante la apreciación directa de sujetos y procesos a partir del estudio del curso natural de los mismos.

Las **tareas de la investigación** que se delimitaron para lograr la culminación exitosa del proceso de construcción de la aplicación fueron:

- Caracterizar el estado del arte de los sistemas de recuperación de videos digitales.
- Describir las herramientas y tecnologías más utilizadas en la actualidad para el desarrollo de software y definir las que se seleccionarán para la realización de la aplicación.
- Seleccionar la metodología de desarrollo a utilizar, la herramienta para el modelado de la aplicación, el lenguaje de programación y el entorno de desarrollo integrado de programación a partir del análisis y comparación de las diversas variantes descritas.
- Confeccionar la documentación que contiene las especificaciones de cada una de las fases transitadas para la construcción y puesta en marcha de la aplicación.
- Implementar el módulo Recuperador del Sistema de Video Vigilancia.

Dentro de los **posibles resultados** que se derivan del cumplimiento de las tareas previstas se tienen:

- Obtención de la documentación UML⁷ para facilitar las actividades de mantenimiento e incorporación de mejoras a la aplicación.
- Obtención del módulo Recuperador del Sistema de Video Vigilancia con el fin de optimizar los procesos de búsqueda y visualización de los videos digitales obtenidos por el sistema.

⁷ Lenguaje Unificado de Modelado o UML por sus siglas en inglés. UML es un lenguaje de modelado estandarizado de uso general en el campo de la ingeniería de software.

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En este capítulo se exponen algunos principios, axiomas o leyes que rigen el funcionamiento del Sistema de Video Vigilancia que actualmente se desarrolla en la facultad y las relaciones entre los procesos que de alguna manera inciden o forman parte de la aplicación informática que se pretende desarrollar e integrar a dicho sistema. Se describe el estado del arte del objeto de estudio y varias de las propuestas alternativas existentes en el mercado que permiten brindar solución a la problemática identificada. Se detallarán además las herramientas y tecnologías que se emplearán en el desarrollo del módulo Recuperador para el Sistema de Video Vigilancia.

1.2. Evolución de los sistemas de recuperación de videos digitales

La vigilancia por video existe hace más 25 años. En sus inicios eran sistemas cien por ciento 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 convirtiéndose en sistemas más complejos y modulares permitiendo la incorporación de numerosas funcionalidades tales como la recuperación de los videos digitales.(Axis)

Analógico	<ul style="list-style-type: none"> • Sistemas de CCTV analógico usando VCR
Parcialmente digital	<ul style="list-style-type: none"> • Sistemas de CCTV analógico usando DVR • Sistemas de CCTV analógico usando DVR de red.
Totalmente digital	<ul style="list-style-type: none"> • Sistemas de video IP que utilizan servidores. • Sistemas de video IP que utilizan servidores y cámaras IP.

Tabla 1. Evolución de los sistemas de video vigilancia

En las últimas décadas, las técnicas de recuperación de medias en los sistemas de vigilancia, han evolucionado en cuanto al tipo de tecnología utilizada. La primera generación de estos sistemas estaba basada en un CCTV analógico (Figura 1). Este está formado por cámaras

analógicas con salida coaxial conectadas a un VCR para grabar que utiliza el mismo tipo de cintas que una grabadora doméstica. En este punto, la recuperación de las medias se realizaba de forma manual por lo que se hacía bien engorroso 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.

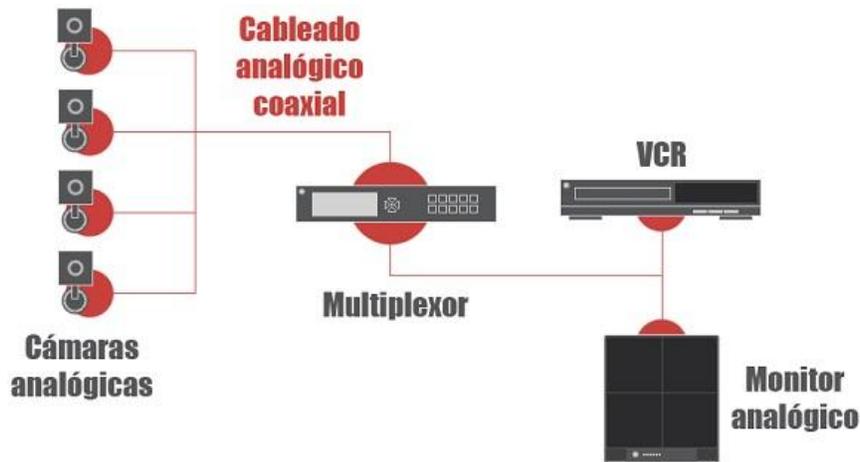


Figura 1. Sistemas de CCTV analógico usando VCR

Con el surgimiento de los grabadores de video digital (DVR) la cinta magnética fue sustituida por el disco duro como medio de almacenamiento pudiendo obtenerse con estas mejoras mayor calidad en la imagen y mayor longevidad de las grabaciones. El empleo de estos dispositivos permitió además 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. En este tipo de sistemas, la búsqueda y visualización de las secuencias de video deseadas seguía siendo una actividad manual.

Posteriormente los sistemas de vigilancia se volvieron parcialmente digitales ya que incluían un DVR IP equipado con un puerto Ethernet para conectividad de red. Como el vídeo se digitalizaba y comprimía en el DVR, se podía transmitir a través de una red informática para que se monitoreara en una PC con una ubicación remota (Figura 2). Pero no fue hasta la creación de los sistemas de vídeo IP que se incorporó a las PC un software de gestión de vídeo con la capacidad de realizar cada una de las tareas que en otro tiempo se realizaban de forma

manual ya que se crearon módulos para garantizar la búsqueda instantánea de medias en la base de datos de la aplicación y la posterior visualización de los videos digitales guardados sin importar la cantidad de cámaras o archivos que se encuentren almacenados en el servidor. En los sistemas de video IP la información del vídeo se transmite de forma continua a través de una red IP y utiliza un servidor de video como elemento clave para migrar del sistema analógico de seguridad a una solución de vídeo IP.

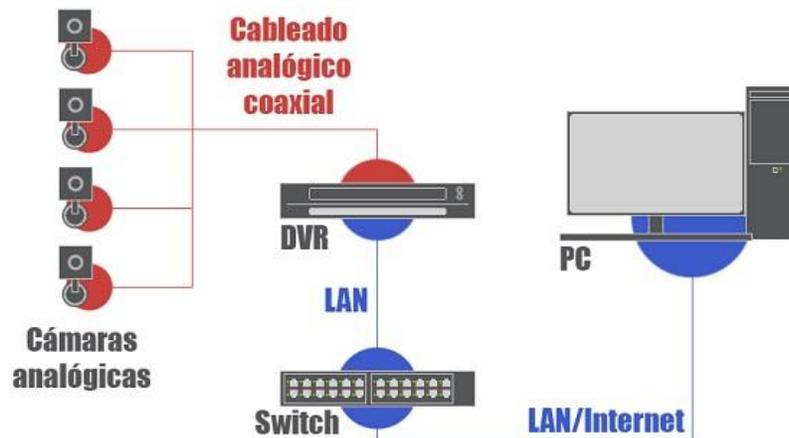


Figura 2. Sistemas de CCTV analógico usando DVR de red

La creación de las cámaras IP trajo consigo el surgimiento de la última generación de los sistemas de video vigilancia sacando el máximo de partido de la tecnología digital. Estas cámaras combinan una cámara y un ordenador en una unidad e incluyen un conector de red. El vídeo se transmite, mediante los conmutadores de red, a través de una red IP y se gestiona en una PC estándar con un sistema de video vigilancia (Figura 3) como los que actualmente producen compañías como *Axis*.

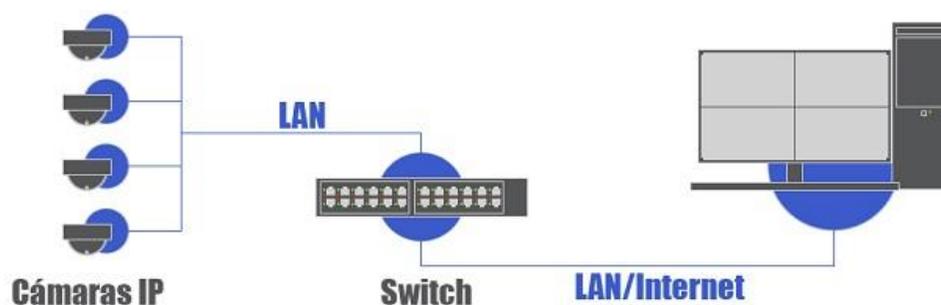


Figura 3. Sistemas de video IP que utilizan servidores y cámaras IP

Actualmente la tendencia de los productores de soluciones de video vigilancia es la de incorporar módulos en los que se agrupan funcionalidades tales como buscar y visualizar los videos digitales almacenados en la base de datos de la aplicación. A continuación se muestran los módulos de recuperación de *Vivotek* y *D-Link* respectivamente:

Vivotek ST7105: Es una nueva generación de software de gestión de video vigilancia que utiliza un sistema de reproducción simultanea de hasta 16 canales, cuenta con un potente motor de búsqueda que es capaz de navegar por las bases de datos de vídeos grabados y además cuenta con funciones avanzadas como la búsqueda, navegación y exportación de medias.(vivotek)



Figura 4. Reproductor de Vivotek ST7105

D-Link: Cuenta con un reproductor avanzado que incorpora una función de búsqueda inteligente que es capaz de detectar eventos importantes con solo señalar una parte de interés en el video. Cuenta además con opciones de detección de movimiento y perdida de foco en cuanto a criterio de búsqueda se trata, lo que lo hace un software muy bueno pero extremadamente caro.



Figura 5: Reproductor del sistema D-Link

1.3. Estado del arte de los sistemas de recuperación de videos digitales

La evolución de la sociedad y la aparición de nuevas tecnologías, en todos los terrenos, han popularizado sistemas que hasta hace poco eran empleados en entornos muy restringidos, bien por sus usos profesionales y aplicaciones muy específicas o simplemente por sus elevados costes. Hoy en día podemos contemplar con normalidad el uso de los CCTV y de los sistemas de video vigilancia en las empresas, escuelas, comunidades, centros comerciales y de salud aumentando cada vez más el número de personas que optan por este tipo de soluciones con el fin de evitar el vandalismo y los hechos delictivos. Las cámaras cumplen una función disuasoria y con su instalación los incidentes de este tipo bajan. No obstante la instalación de una cámara debe estar justificada por una razón de seguridad, no debe ser una decisión tomada a la ligera y se tiene que instalar de la forma menos intrusiva en la privacidad de las personas.

A nivel internacional contamos con una gran cantidad de sistemas de video vigilancia que en su mayoría son privativos y se distribuyen conjuntamente con el hardware, es decir, que las empresas encargadas de la producción de cámaras IP comercializan un software capaz de interactuar únicamente con sus productos, lo que representa un gran inconveniente pues esto aumenta el costo de adquisición del sistema. Entre estas empresas se tienen a:

Axis: Es la 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 a sistemas de video vigilancia que ha desarrollado esta empresa es el *AXIS Camera Station*

capaz de proporcionar funciones de supervisión de video, grabación y gestión de eventos, también cuenta con múltiples funciones de búsqueda de eventos grabados.(Axis)



Figura 6. Sistema Axis Camera Station

Panasonic: Junto a Axis es una de las empresas de mayor distribución de cámaras IP que se caracteriza por vender los kit de series de cámara con su respectivo software de vigilancia por lo que ningún software de un kit es compatible con otro. Tiene entre sus productos cámaras resistentes que soportan la variación de las temperaturas externas con visión de 360° y que permiten un control las 24 horas.



Figura 7. Sistema para kit de cámaras Panasonic

ZoneMinder: Es una solución de software libre, que permite implementar un sistema de monitoreo de cámaras de vigilancia. Es un sistema híbrido basado en web, presenta una interfaz para cualquier dispositivo y no requiere ni descargar ni instalar nada para el equipo cliente. El sistema cuenta con las mismas funcionalidades que otros productos como la monitorización la grabación de videos y la posterior reproducción de las mismas. Este sistema presenta incompatibilidad con algunos dispositivos, no tiene soporte de audio no es multiplataforma, trabaja solo en sistemas GNU/Linux y está pensado para usuarios profesionales por lo que presenta una interfaz para nada amigable.(aura, 2010)

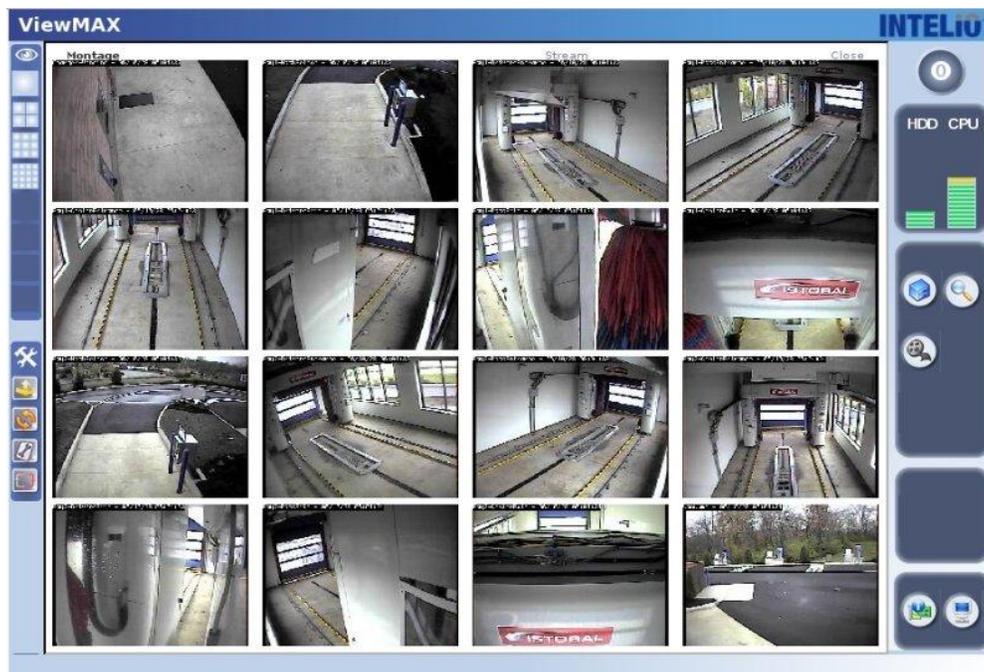


Figura 8. Sistema ZoneMinder

En nuestro país hay sectores que necesitan de las comodidades que brinda la instalación de un sistema de video vigilancia por lo que se ha incursionado en el desarrollo de este tipo de soluciones por parte de la empresa de desarrollo de tecnologías y sistemas *DATYS* la cual cuenta con un sistema de video vigilancia de última generación llamado *XYMA SAFE VISION* que es un software de video vigilancia profesional basado en tecnología IP con un alto grado de modularidad. Es además adaptable, flexible y escalable y permite el uso de tecnologías analógicas. Permite administrar, monitorear, grabar, revisar, configurar alertas y alarmas y obtener reportes, para cualquier solución de video vigilancia que se requiera implementar. Este software de manufactura nacional es un gran avance en el desarrollo de este tipo de soluciones pero tiene como inconveniente que trabaja sobre software propietario además de tener un alto

precio en el mercado, por lo que en la facultad 6 de la UCI se trabaja en la realización de un sistema de video vigilancia que cuente con un módulo de recuperación de videos digitales que sea multiplataforma.

1.4. Tecnologías, herramientas y metodologías a utilizar para el desarrollo de la solución.

1.4.1. Tecnologías

Las tecnologías a utilizar a la hora de desarrollar cualquier aplicación están estrechamente relacionadas con el lenguaje de programación a utilizar para el proceso de implementación, entre ellos se encuentran:

- **C#:** Es el lenguaje insignia de la plataforma *.net*, cuenta con una extensa librería base para el desarrollo de todo tipo de aplicaciones. *C#* es un lenguaje de programación simple pero eficaz, solo que para su uso adecuado se deberá tener instalado una versión reciente de *.net* por lo que lo hace un lenguaje muy poco útil en cuanto al trabajo multiplataforma.
- **Java:** Es un lenguaje multiplataforma que contiene una rica colección de clases llamadas *librería de clases de Java* que cuenta con varios miles de ellas y 10 mil métodos que proveen un mejor manejo de archivos, el trabajo con interfaces gráficas la comunicación de datos, el acceso a internet y el acceso a bases de datos. Pero para aplicaciones de envergadura se requiere de *frameworks*⁸ externos y las aplicaciones resultantes requieren de gran cantidad de memoria en las computadoras clientes.
- **C++:** Es un lenguaje increíblemente versátil. Con él pueden programarse desde los programas más simples hasta los más complicados. Además es portable, es decir, un programa con el código escrito en C++, se podrá compilar en cualquier sistema operativo o sistema informático si necesidad de cambiar casi el código fuente. Otras de las grandes ventajas del C++, es que es un lenguaje multi-nivel, es decir, puede ser usado tanto para programar directamente el hardware (dependiendo del sistema operativo), como para crear aplicaciones tipo Windows definidas todas por poseer una misma interfaz.(Curso C++)

⁸Librería de código (posiblemente ya compilado) que sirve como base para el desarrollo de un tipo determinado de aplicaciones.

1.4.2. Herramientas

1.4.2.1. Ingeniería y documentación

Más conocidas por herramientas *CASE*⁹. Las herramientas más reconocidas en este campo son:

- **RationalSuite:** Es una solución integrada y completa para todo el ciclo de vida de desarrollo de software que es capaz de acelerar el desarrollo mediante la modelación visual, la generación de código y las capacidades de ingeniería inversa. Es bastante cara y poco intuitiva de trabajar.
- **Visual Paradigm:** Es una *Suite* totalmente multiplataforma que soporta el ciclo de vida completo del desarrollo del software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue, además de que ayuda a construir aplicaciones de manera más rápida y económica, solo que requiere bastante memoria *RAM*¹⁰ debido a que está desarrollada en *JAVA*¹¹.
- **Enterprise Architect:** Es una herramienta progresiva que cubre todos los aspectos del ciclo de desarrollo, proporcionando una trazabilidad completa desde la fase inicial del diseño a través del despliegue y mantenimiento. También provee soporte para pruebas, mantenimiento y control de cambio(ENTERPRISE ARCHITECT, 2007). Entre las principales características que este brinda se pueden mencionar:
 - Crea elementos del modelo *UML* para un amplio alcance de objetivos.
 - Ubica esos elementos en diagramas y paquetes.
 - Documenta los elementos que se han creado.
 - Genera el código fuente para el software que se está construyendo.
 - Realiza ingeniería directa e inversa de código en lenguajes como *ActionScript*, *C++*, *C#*, *Java*, *PHP*, entre otros.

⁹ CASE: Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador.

¹⁰ RAM: Random Access Memory. Memoria de Acceso Aleatorio. Tipo de memoria utilizada por las computadoras.

¹¹ JAVA: Lenguaje de desarrollo multiplataforma.

- Soporta todos los diagramas y modelos del UML. Puede modelar procesos de negocio, sitios web, interfaces de usuario, redes, configuraciones de hardware, mensajes y más.
- Estima el tamaño del proyecto en esfuerzo de trabajo en horas. Captura y traza requisitos, recursos, planes de prueba, solicitudes de cambio y defectos.

1.4.2.2. IDE de desarrollo

Los *IDE*¹² se deben seleccionar en dependencia de los lenguajes en que se piensa desarrollar la aplicación, debido a que no todos los *IDE* soportan los lenguajes existentes. Como se ha seleccionado C++ como lenguaje de programación entre los posibles candidatos entre los distintos *IDE* de desarrollo se encuentran:

- **KDevelop:** El proyecto *KDevelop* iniciado en 1998, pretendía por ese entonces diseñar un entorno de desarrollo integrado fácil de usar para C/C++ en *Unix*. Desde entonces está disponible públicamente bajo licencia *GPL*, y soporta *KDE/Qt*, *GNOME*, *C* y *C++*. El equipo de desarrollo de *KDevelop* ha anunciado la disponibilidad de la versión 4.0 del *IDE* oficial del proyecto *KDE*. Tiene como desventaja principal que su entorno gráfico es algo pobre y solo corre bajo plataformas *GNU/LINUX*.
- **Eclipse:** *Eclipse* fue desarrollado originalmente por *IBM* como el sucesor de su familia de herramientas para *VisualAge*. *Eclipse* es ahora desarrollado por la *Fundación Eclipse*, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto. Los lenguajes con proyectos más importantes son: *Java*, *C/C++*, *PHP* y *Cobol*. Sus mayores ventajas radican en su gran comunidad de desarrollo que lo ubican como el mejor *IDE Java*.
- **QTCreator:** Entre las funcionalidades que ofrece se puede mencionar que cuenta con:
 - Editor de código avanzado con soporte para C++.
 - Herramientas para la rápida navegación del código.
 - Resaltado de sintaxis y auto-completado de código.

¹² Son los entornos de desarrollo integrado (*Integrated Developed Environment* por sus siglas en inglés) usados por los programadores para realizar aplicaciones.

- Control estático de código y estilo a medida que se escribe.
- Ayuda sensitiva al contexto.
- Depurador visual (*visual debugger*) para C++, es consciente de la estructura de muchas clases de Qt, lo que aumenta la capacidad de mostrar los datos de Qt con claridad.
- Muestra la información en bruto procedente del *GNU Project Debugger (GDB)* de una manera clara y concisa.
- Interrupción de la ejecución del programa.
- Ejecución línea por línea o instrucción a instrucción.
- Puntos de interrupción (*breakpoints*).
- Examina el contenido de llamadas a la pila (*stack*), los observadores y de la variables locales y globales.

1.4.3. Metodologías

XP: La *Programación Extrema* fue desarrollada por *Kent Beck* y es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Surgió como respuesta y posible solución a los problemas derivados del cambio en los requerimientos, se plantea como una metodología a emplear en proyectos de riesgo por lo que su uso aumenta la productividad. El ciclo de vida ideal de la metodología cuenta con 6 fases fundamentales que son:

- **Exploración:** En esta fase el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán, también se prueba la tecnología y se construye un prototipo de arquitectura de sistema.
- **Planificación de la Entrega (*Release*):** En esta fase se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Los programadores son los encargados de establecer las estimaciones de esfuerzos asociados a la implementación utilizando como medida el punto lo que equivale a una semana ideal de programación.
- **Iteraciones:** En esta fase se encuentran las iteraciones de no más de tres semanas sobre el sistema antes de ser entregado. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto y al final de la última de las iteraciones el sistema estará listo para entrar en producción.

- **Producción:** Las pruebas adicionales y revisiones de rendimiento son realizadas en esta fase antes de que el sistema sea trasladado al entorno del cliente por lo que se deben tomar decisiones sobre la inclusión o no de nuevas características sobre la versión actual del producto. Las ideas propuestas son documentadas para su posterior implementación.
- **Mantenimiento:** El proyecto *XP*, mientras la primera versión se encuentra en producción, debe al mismo tiempo que desarrolla nuevas iteraciones mantener el sistema en funcionamiento. Esta fase puede requerir la inclusión de nuevo personal dentro del equipo y cambios en su estructura.
- **Muerte del Proyecto:** Cuando se satisfagan las necesidades del cliente en cuanto a rendimiento y confiabilidad del sistema se genera la documentación y no se realizan más cambios en la arquitectura. A esta fase se puede arribar también cuando el sistema no genere los beneficios esperados por el cliente o no haya presupuesto para mantenerlo.

Las características fundamentales del método son:

- Desarrollo iterativo e incremental: pequeñas mejoras, una tras otras.
- Pruebas unitarias¹³ continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión¹⁴.
- Programación en Pareja: se recomienda que las tareas de desarrollo sean llevadas a cabo por dos personas en el mismo puesto ya que así el código puede ser discutido y revisado mientras se escribe aumentando la calidad del mismo.
- Frecuente integración del equipo de programación con el cliente o usuario.
- Corrección de todos los errores antes de agregar nuevas funcionalidades.
- Refactorización del Código: se basa fundamentalmente en reescribir ciertas partes del código para aumentar la legibilidad y mantenibilidad pero sin modificar su comportamiento.
- Propiedad del código compartida: Este método promueve que todo el personal pueda corregir y extender cualquier parte del proyecto.

¹³ Prueba Unitaria: Es una forma de probar el correcto funcionamiento de un módulo de código, mediante casos de prueba extremos.

¹⁴Prueba de Regresión: Tipo de pruebas de software que intentan descubrir las causas de nuevos errores, carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software.

- Simplicidad del código: Según la programación extrema es más sencillo hacer algo simple y posteriormente modificarlo con un poco de trabajo extra si se requiere, que realizar algo complicado y quizás nunca utilizarlo. La simplicidad y la comunicación son extraordinariamente complementarias.

RUP: El *Proceso Unificado de Desarrollo* es un proceso de desarrollo de software que junto con el lenguaje *UML* constituye la metodología estándar de mayor utilidad para el análisis, implementación y documentación de sistemas orientados a objetos. También es una plataforma flexible de procesos de desarrollo de software que provee ayuda con guías consistentes y personalizadas de procesos para todo el equipo del proyecto. A diferencia de otras metodologías *RUP* es capaz de hacer que el proceso sea práctico con bases de conocimientos y guías para ayudar el despegue de la planificación del proyecto, integrar rápidamente a los miembros del equipo y poner en marcha el proceso personalizado.

Una de las mejores prácticas de RUP es la posibilidad de desarrollar iterativamente ya que organiza los proyectos en términos de disciplina y fases, consistiendo cada una de ellas en una o más iteraciones. La aproximación iterativa ayuda a mitigar los riesgos de forma temprana y continua, con un proceso demostrable y constantes *releases ejecutables*. Estas fases son:

- **Concepción:** se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.
- **Elaboración:** se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- **Construcción:** se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.
- **Transición:** se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

Para cada una de estas fases definen los siguientes flujos de trabajo:

1. Modelado del negocio.
2. Requerimientos.
3. Análisis y diseño.
4. Implementación.

5. Prueba.
6. Despliegue.
7. Configuración y manejo de cambios.
8. Administración del proyecto.
9. Gestión del entorno.

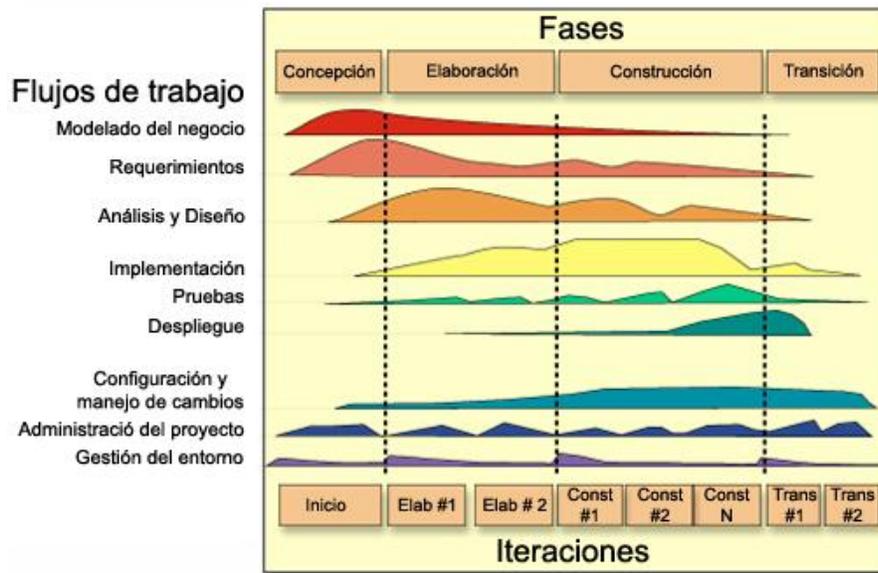


Figura 9. Descripción de la composición de RUP

Selección: Por todo lo antes expuesto se decidió que la metodología de desarrollo de software que se empleará en la planificación de la construcción de la aplicación será *RUP* y por las características con las que cuenta el módulo que se pretende realizar, se ha decidido escoger como lenguaje de programación *C++*, se elige el *QTCreator* como *IDE* de desarrollo a emplear para la realización del módulo Recuperador para el sistema de video vigilancia y de las herramientas *CASE* anteriormente descritas se optó por la selección del *Enterprise Architect* para la modelación de la aplicación.

1.5. Conclusiones del capítulo

A partir de la investigación realizada para la obtención de las características particulares de cada una de las actividades que en su conjunto conforman el sistema de videovigilancia *Suria SIV* se obtuvo como resultado una descripción detallada de las mismas que permitieron definir,

de manera concreta, cada uno de los problemas que limitan rapidez en cuanto a las operaciones a realizar sobre los videos almacenados que implican una carga de trabajo innecesaria para el personal dedicado al control del sistema.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

2.1. Modelo de Dominio

Los modelos de dominio pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir.(synergix, 2008)

2.1.1. Conceptos Fundamentales

- **Operador:** Persona encargada de interactuar con el sistema.
- **Agente Externo (Gestor):** Sistema encargado de proveer la información necesaria para el funcionamiento de los agentes autónomos.
- **Sistema Recuperador:** Sistema encargado de realizar la búsqueda y la reproducción de los videos.

2.1.2. Diagrama del Modelo de Dominio

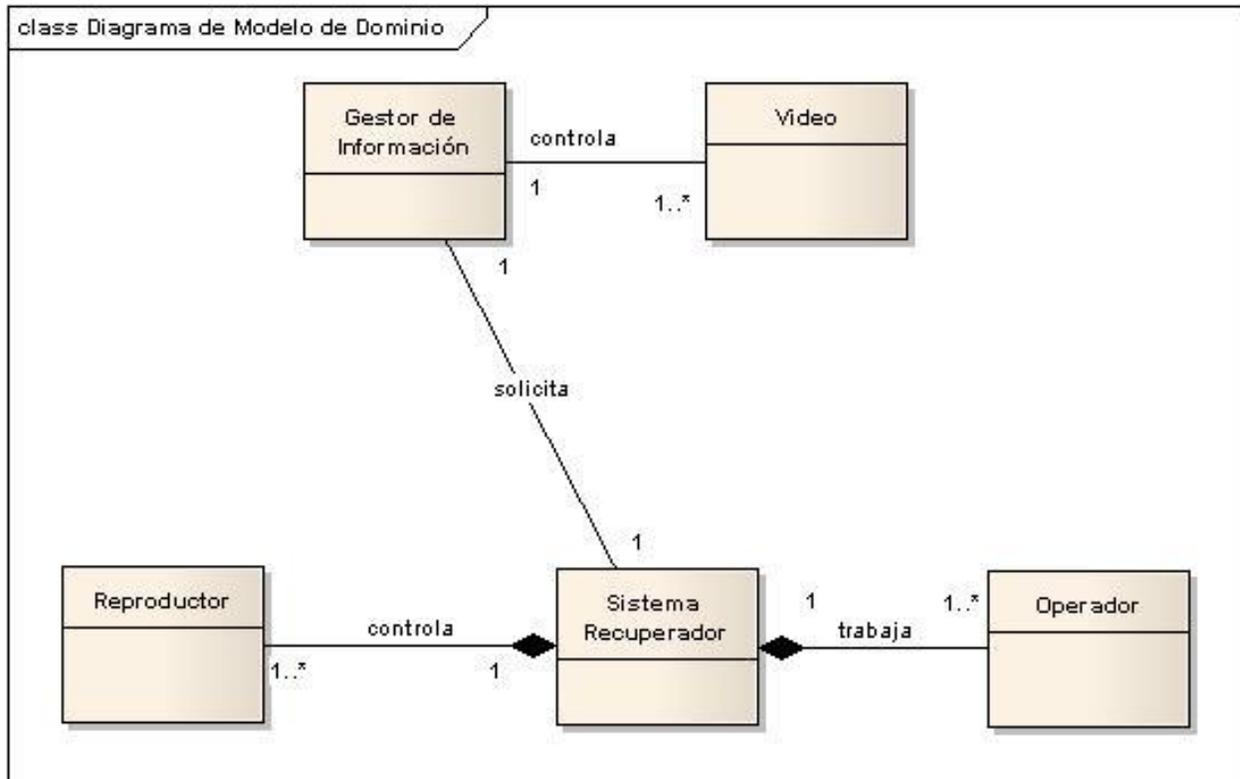


Figura 10. Modelo de Dominio

2.2. Propuesta de Sistema

La solución que se plantea es la de desarrollar un Módulo Recuperador multiplataforma que permita realizar las tareas de búsqueda y visualización de los videos digitales obtenidos por el Sistema de Video Vigilancia, lo cual evitará el gasto innecesario de tiempo en las operaciones realizadas sobre los archivos, además de tener la capacidad de poder ser utilizado sobre diferentes sistemas operativos como Windows o Linux.

2.3. Requisitos Funcionales del Sistema

Esta sección describe los requisitos funcionales del Subsistema de Sistema de Video Vigilancia expresado en lenguaje natural y de forma clara.

2.3.1. Requerimientos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir.

- **RF 1 Autenticar Usuario.**

Con este caso de uso el sistema permite al usuario que se autentique y le da permisos y privilegios según el rol que tenga en el sistema.

- **RF 2 Buscar videos grabados.**

Este requerimiento permite hacer búsqueda de los videos que se grabaron con anterioridad de acuerdo a criterios de búsquedas.

- **RF 3 Visualizar videos grabados.**

Este requerimiento le da la posibilidad al usuario de visualizar uno o varios flujos de videos a la vez.

- **RF 4 Exportar videos.**

Este requerimiento permite exportar los videos en un formato común.

- **RF 5 Configurar conexión.**

Este requerimiento le permite al usuario configurar los parámetros de conexión del Recuperador con el Gestor.

2.4. Requerimientos no Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Son características que hacen al producto atractivo, usable, rápido o confiable. Estos requerimientos se agrupan en varias categorías:

2.4.1. Requerimientos de Usabilidad

- **RNU 1** La interfaz debe ser configurable para lograr la comodidad del usuario.
- **RNU 2** El Cliente¹⁵ debe tener una interfaz gráfica, visualmente atractiva para el usuario.
- **RNU 3** El usuario debe poder configurar el sistema, sin necesidad de acceder manualmente a los ficheros de configuración.

¹⁵ Cliente: Refiriéndose a la aplicación cliente del Recuperador.

- **RNU 4** El sistema deberá estar bajo las normas establecidas de la norma ISO 9241-11, la cual define los parámetros internacionales de usabilidad de cualquier software.

2.4.2. Requerimientos de Fiabilidad

- **RNF 5** El sistema debe estar disponible todas las horas del día. Se le debe dar mantenimiento al servidor cada 15 días como máximo.
- **RNF 6** Los errores más comunes que afectan directamente la fiabilidad del sistema se han identificado y categorizado de la siguiente forma:

Menores

Significativos

Críticos.

2.4.3. Requerimientos de Eficiencia

- **RNE 7** El sistema debe ser eficiente a las peticiones realizadas en cada momento, el flujo de trabajo que sigue la aplicación no permite el fallo de ninguna de las partes, influye de manera drástica sobre el próximo paso.
- **RNE 8** El tiempo de respuesta promedio de las transacciones u operaciones realizadas en el sistema no debe exceder un tiempo máximo de 5 segundos.
- **RNE 9** El sistema constituye la integración de los subsistemas de la solución de Video Vigilancia por lo que debe ser lo más eficiente posible, poseer velocidad para el procesamiento de la información y cortos tiempos de respuesta.

2.4.4. Requerimiento de Soporte

Las opciones de instalación y mantenimiento del sistema deben ser fáciles, además debe ser adaptable a cualquier plataforma libre. Los instaladores deben incluir todas las librerías que necesite el sistema para estar corriendo.

- **RNSO 10** El sistema permitirá la modificación o agregarle módulos cuando sea necesario, asegurando su extensibilidad y lograr mejores prestaciones.

- **RNSO 11** La implementación del sistema seguirá las normas de codificación establecidas¹⁶.

2.4.5. Requerimiento de Funcionamiento

- **RNFO 12** Debe tener instaladas librerías de QT 4.7 o superior.

2.5. Definición de Casos de Uso

2.5.1. Definición de los actores.

Actor	Descripción
Operador	Este actor es el que va a realizar todas las operaciones sobre el modulo recuperador desde visualizar, filtrar y exportar videos

Tabla 2. Actor del Sistema

2.5.2. Especificación de los Casos de Uso del Sistema(CUS)

2.5.2.1. Descripción del CUS Autenticar usuario

Caso de Uso:	AutenticarUsuario
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el actor introduce su usuario y contraseña para acceder al sistema, los mismos se verifican contra la base de datos. Finaliza cuando se habilita la entrada al usuario con los

¹⁶Soporte\Aseguramiento de la Calidad\040 Lineamientos de Calidad (IPL 3101-2008) v1.0.pdf

	permisos asignados a este o se deniega su acceso.
Precondiciones:	
Referencias:	RF 1
Prioridad:	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El usuario entra el usuario y contraseña y oprime el botón aceptar.	2. El sistema registra los datos entrados por el usuario y le muestra la interfaz según los privilegios del usuario.

Prototipo de Interfaz

Flujos Alternos

Acción del Actor	Respuesta del Sistema
1.1 El usuario entra el usuario y contraseña y oprime el botón aceptar.	2.1. Si el usuario entra algún dato mal, el sistema le informa y le muestra nuevamente la interfaz de autenticación.
Poscondiciones	Luego de autenticado el usuario en el sistema se le da los privilegios y permisos que tenga en la aplicación

Tabla 3.CUS_AutenticarUsuario

2.5.2.2. Descripción del CUS Exportar videos

Caso de Uso:	Exportar videos
Actores:	Operador, Administrador.
Resumen:	El caso de uso se inicia cuando se introduce la dirección donde se desea salvar el video, esta es verificada; el caso de uso finaliza mostrando un mensaje de aviso correspondiente.

Precondiciones:	El usuario debe estar autenticado previamente.
Referencias	RF 4
Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Exportar.	2. El sistema le da la posibilidad de exportar el video.

Prototipo de Interfaz

Flujos Alternos

Acción del Actor	Respuesta del Sistema

Tabla 4.CUS_ExportarVideo

2.5.2.3. Descripción del CUS Buscar videos

Caso de Uso:	Buscar videos
Actores:	Operador
Resumen:	El caso de uso se inicia cuando se introducen las fechas de inicio y fin de grabación del video o la cámara por la que fue grabada la secuencia de datos, estos son verificados; el caso de uso finaliza mostrando la interfaz con los videos correspondientes.
Precondiciones:	El usuario debe autenticarse en el sistema.
Referencias	RF 2
Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El operador selecciona la opción buscar videos. 3. El operador escoge los criterios de búsqueda de los videos.	2. El sistema muestra una interfaz para que seleccione por los criterios que lo va a buscar. 4. El sistema muestra una interfaz con los videos.
Flujos Alternos	
Prototipo de Interfaz	
Poscondiciones	

Tabla 5.CUS_BuscarVideo

2.5.2.4. Descripción del CUS Visualizar video

Caso de Uso:	Visualizar video.
Actores:	Operador
Resumen:	El caso de uso se inicia cuando el actor selecciona en el reproductor el video a visualizar, el video se carga; el caso de uso finaliza habilitando la interfaz con las operaciones correspondientes.
Precondiciones:	El usuario debe autenticarse en el sistema.
Referencias	RF 3
Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El operador selecciona la opción Visualizar video. 3. El operador escoge el flujo de video a visualizar.	2. El sistema muestra una interfaz para que seleccione el flujo de video que desea visualizar. 4. El sistema muestra una interfaz con el

	flujo de video.
Prototipo de Interfaz	
Poscondiciones	

Tabla 6.CUS_VisualizarVideo

2.5.3. Diagrama de Casos de Uso

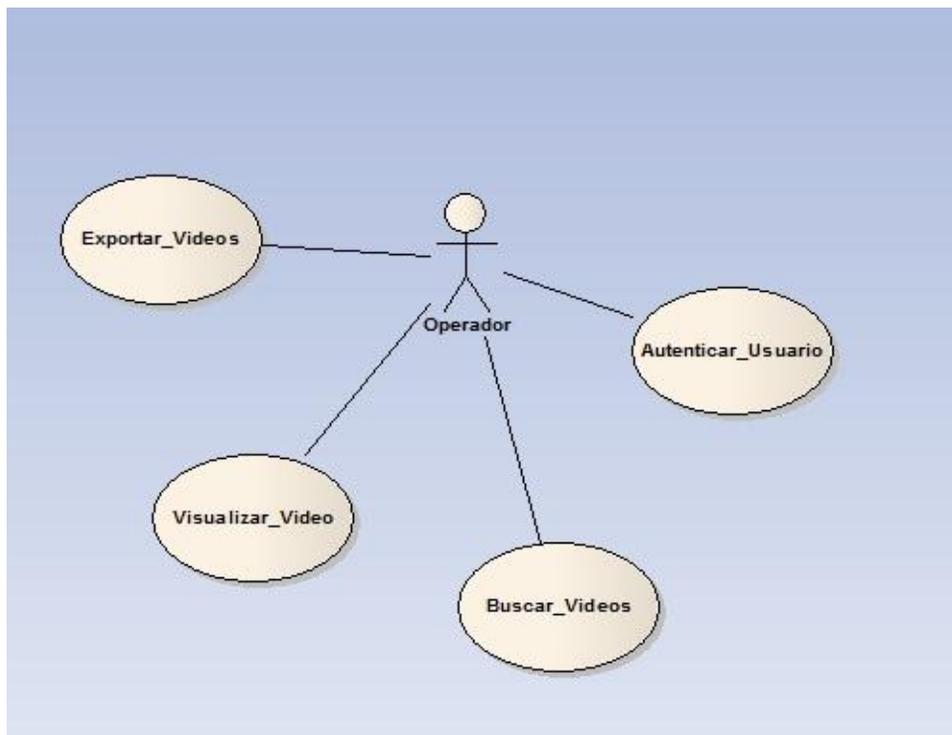


Figura 11.Diagrama de Casos de Uso

2.5.4. Conclusiones del capítulo.

En este capítulo se establecieron las características principales del sistema así como el proceso para su funcionamiento. Se muestra el modelo del dominio, conjuntamente con la

especificación de los requisitos funcionales y no funcionales. Además se elaboró el modelo de caso de usos del sistema.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA

3.1. Arquitectura

- **Arquitectura en Pizarra**

La elección de este estilo de arquitectura se basa en que el mundo del video vigilancia está en constante movimiento y evolución. Con frecuencia surgen nuevas tecnologías que son adoptadas automáticamente en la realización de estos sistemas, o se desarrollan algunas específicamente para ellos. Estos cambios van desde hardwares con funcionalidades novedosas hasta nuevas características o prestaciones del software que todos se apresuran a incorporar para no quedar en desventaja competitiva. La Arquitectura en Pizarra brinda una sólida estructura, que es flexible al cambio, por lo cual es una respuesta adecuada a las necesidades del sistema.

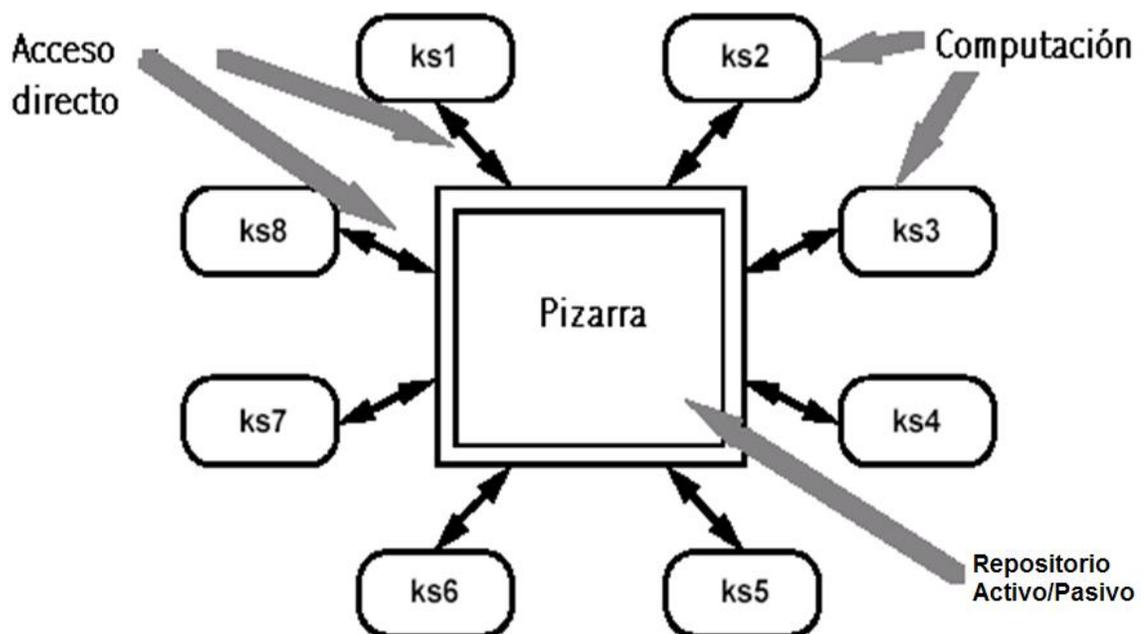


Figura 12. Arquitectura en Pizarra

El Gestor es la pizarra del sistema, a la cual se conectan cada uno de los restantes módulos, los cuales no tienen necesidad de conocer de la existencia de los demás, simplemente obtienen información de la pizarra, y también postean información y eventos, los cuales pueden obtener los demás agentes y responder a ellos. Esto hace

posible la obtención de funcionalidades complejas mediante la coordinación de las entradas y las salidas de un determinado número de agentes autónomos.

- **Patrón Modelo-Vista-Controlador**

En el módulo Recuperador se utiliza el estilo Modelo-Vista-Controlador (MVC). Este afronta una de las consecuencias de los cambios frecuentes en los soportes de hardware y de añadir nuevas funcionalidades, que es la de constantes modificaciones en la interfaz de usuario. Se separa los módulos en 3 componentes fundamentales, el Modelo, la

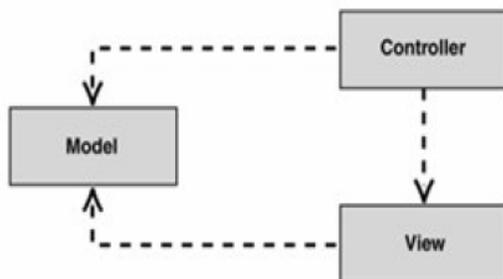


Figura 13. Patrón *Modelo-Vista-Controlador*

Vista y el Controlador. El Modelo es inmutable a los cambios en la interfaz de usuario, se encarga de toda la gestión interna del negocio, la Vista se encarga de presentar los datos y de interactuar con el usuario, mientras que el controlador maneja todos los cambios de estado de la Vista. Mediante el uso de este patrón el Reproductor por ejemplo, es capaz de utilizar diferentes interfaces de usuario

manteniendo la misma lógica de negocio.

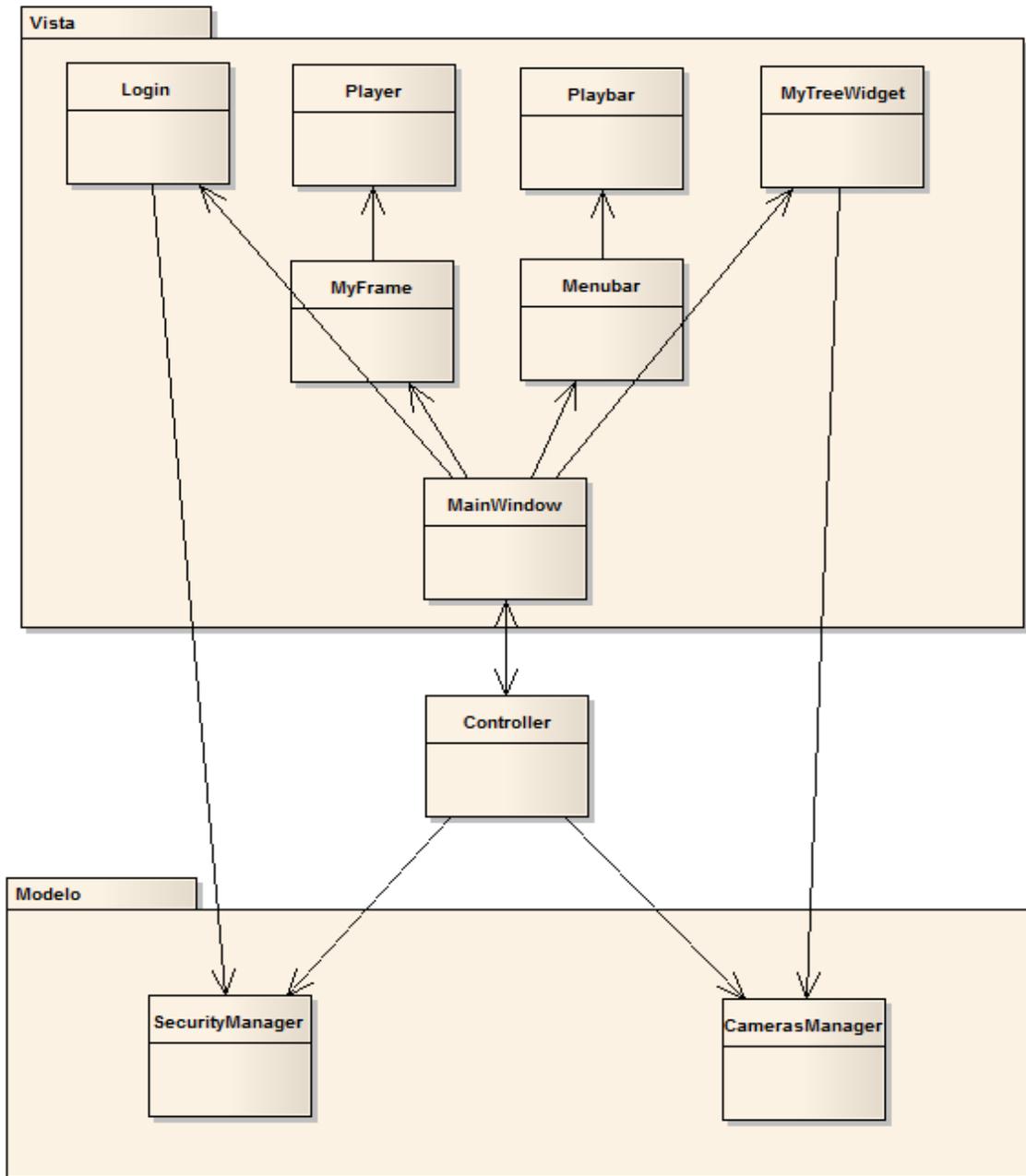


Figura 14.Arquitectura en el módulo Recuperador

3.2. Modelo de Análisis.

El análisis es una abstracción, es un bosquejo del diseño de la aplicación el cual define una estructura para modelar el sistema.

3.2.1. Diagrama de clases de análisis.

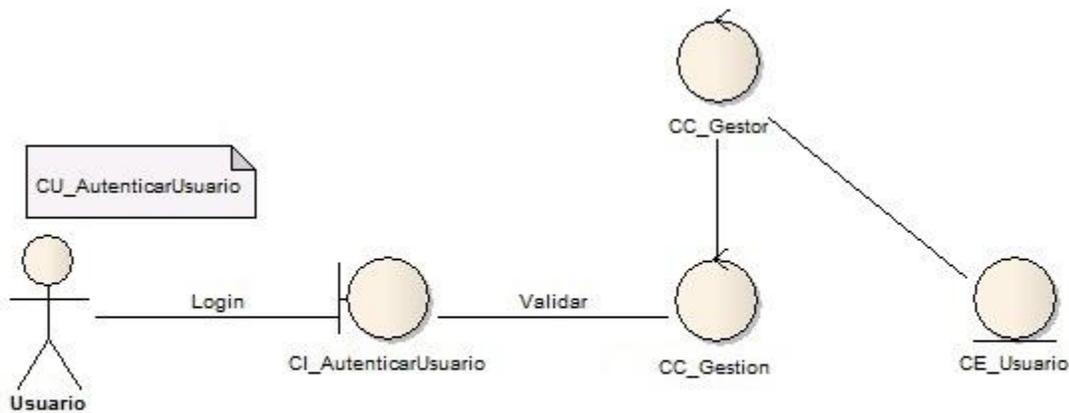


Figura 15. CU Autenticar usuario

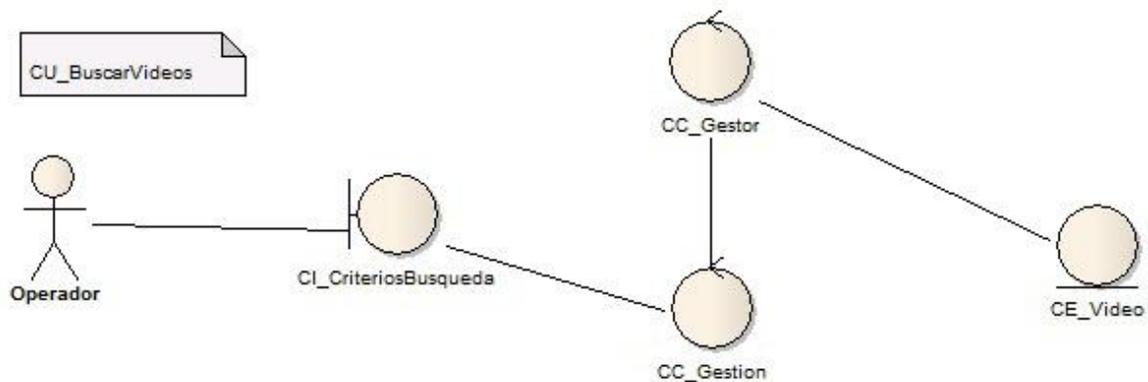


Figura 16. CU Buscar videos

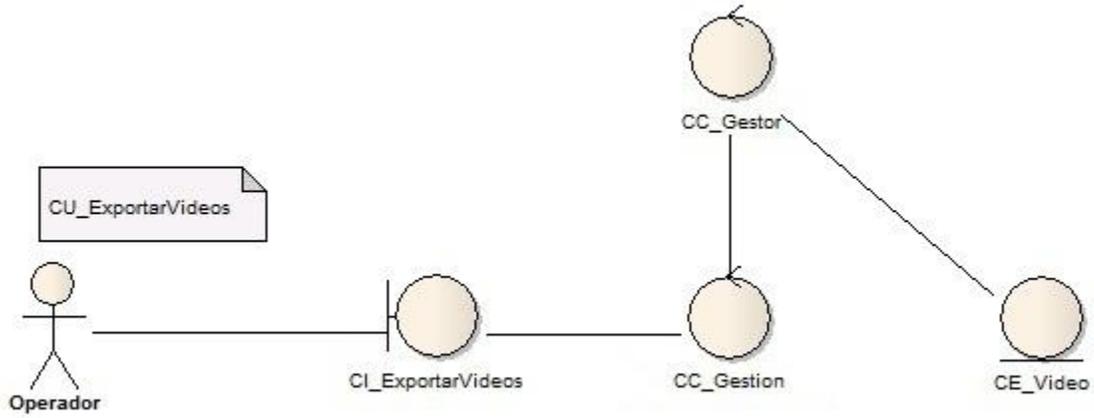


Figura 17.CU Exportar videos

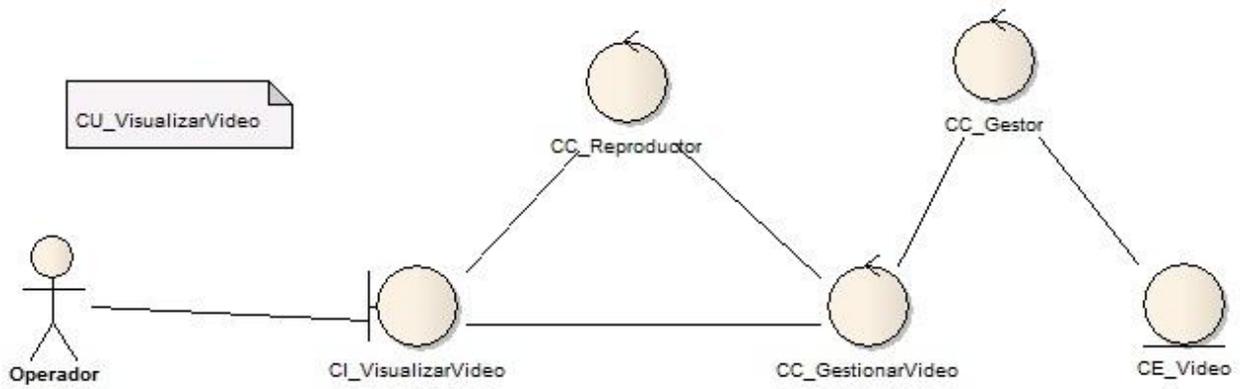


Figura 18.CU Visualizar video

3.3. Modelo de Diseño.

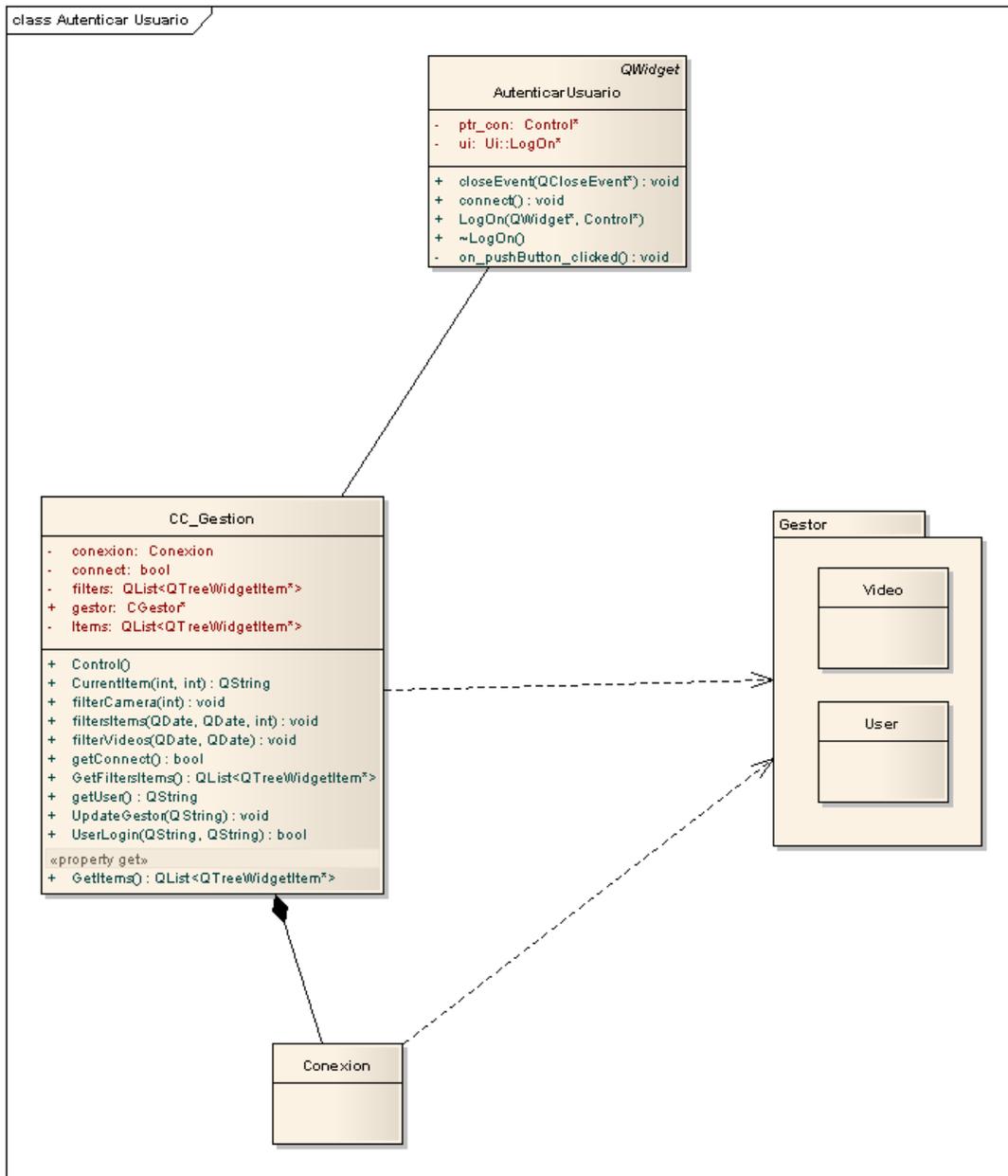


Figura 19. Diagrama de clases del diseño *CU_AutenticarUsuario*

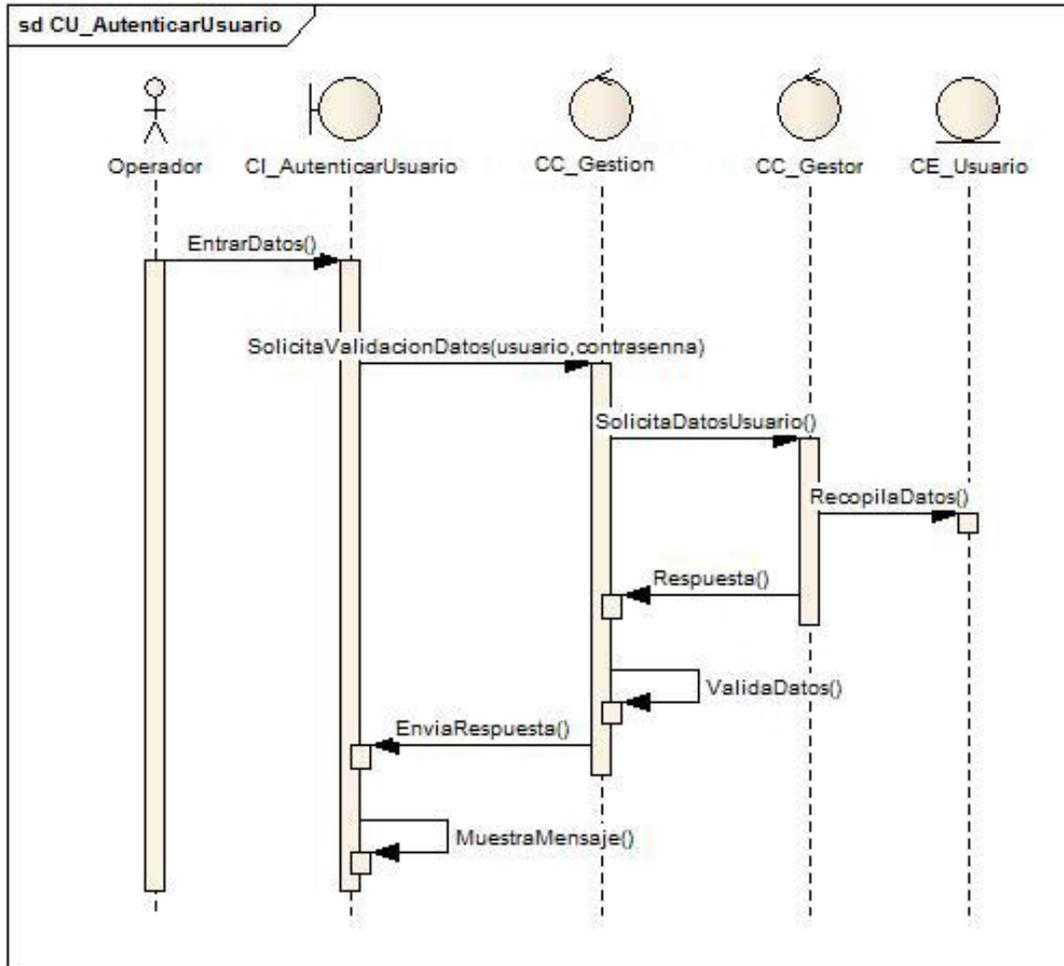


Figura 20. Diagrama de secuencia *CU_AutenticarUsuario*

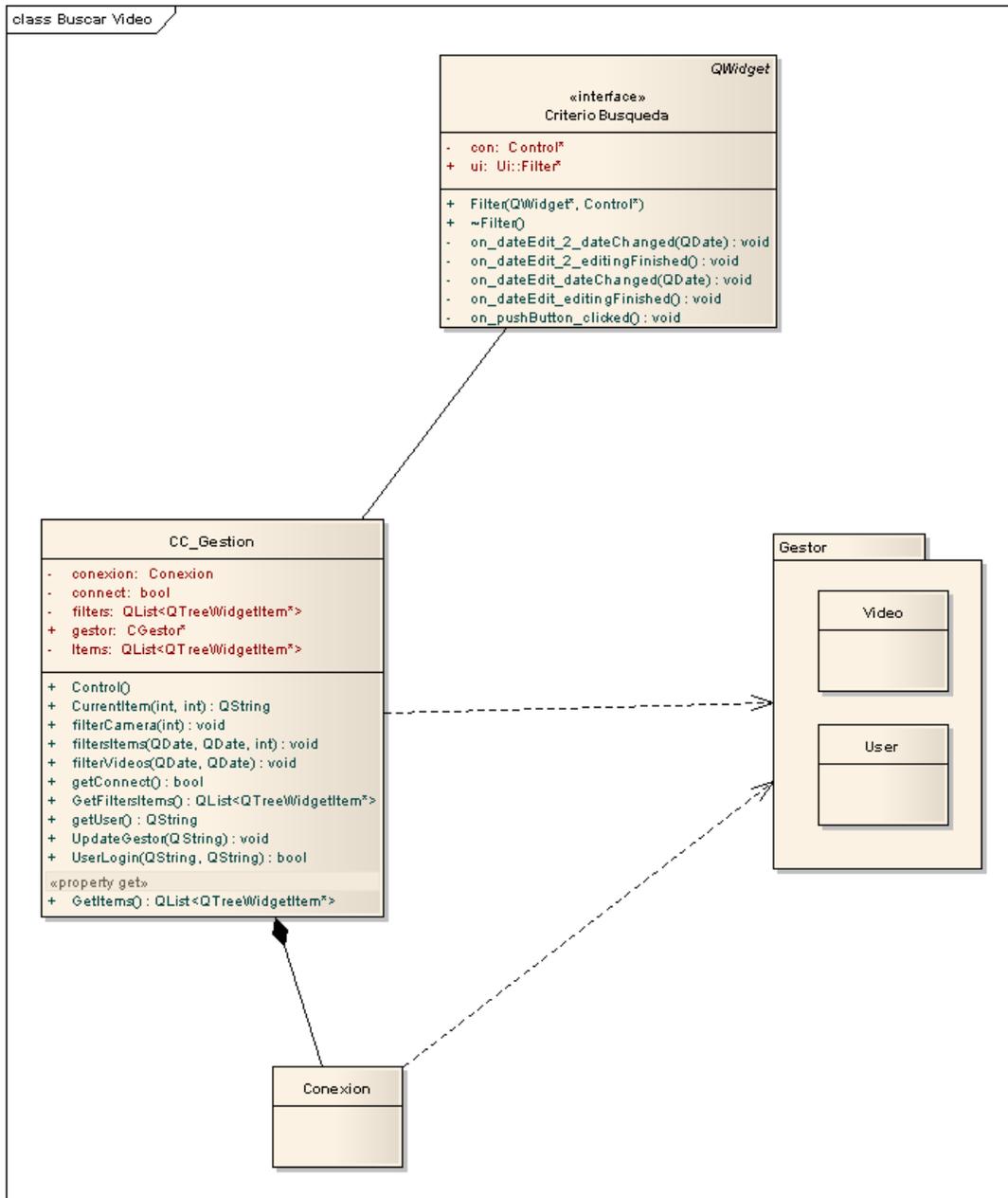


Figura 21. Diagrama de clases del diseño *CU_BuscarVideo*

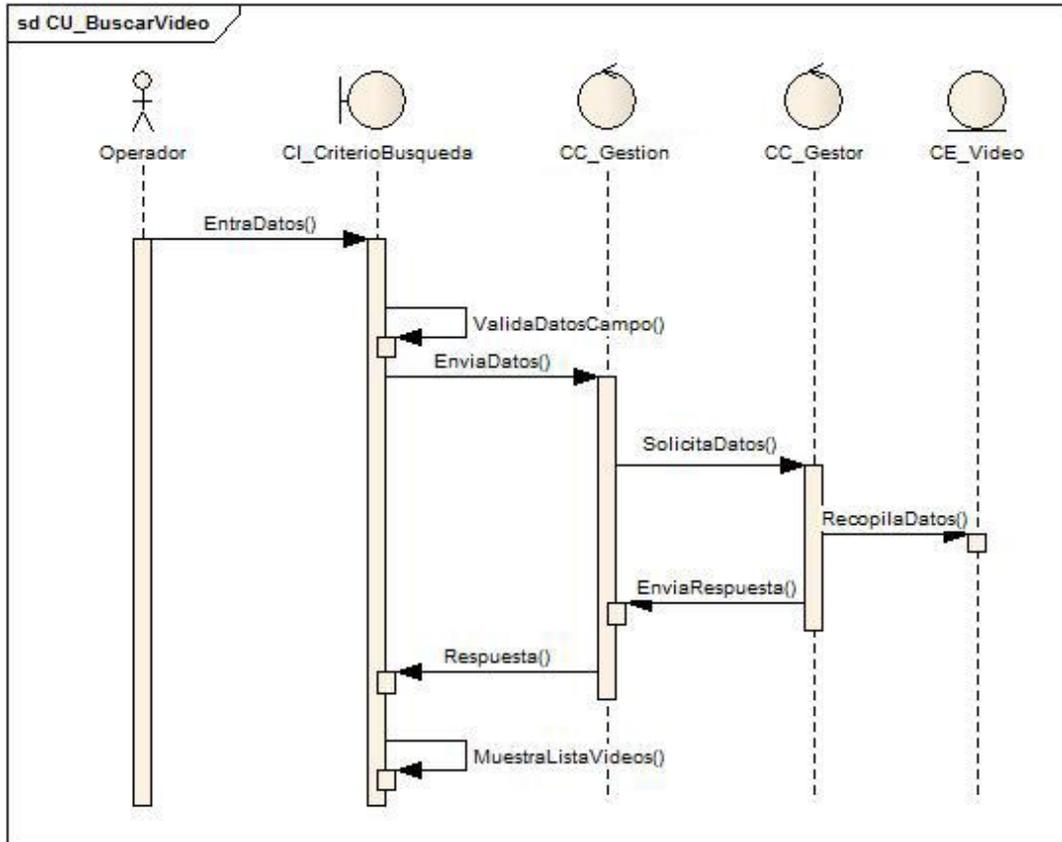


Figura 22. Diagrama de secuencia *CU_BuscarVideo*

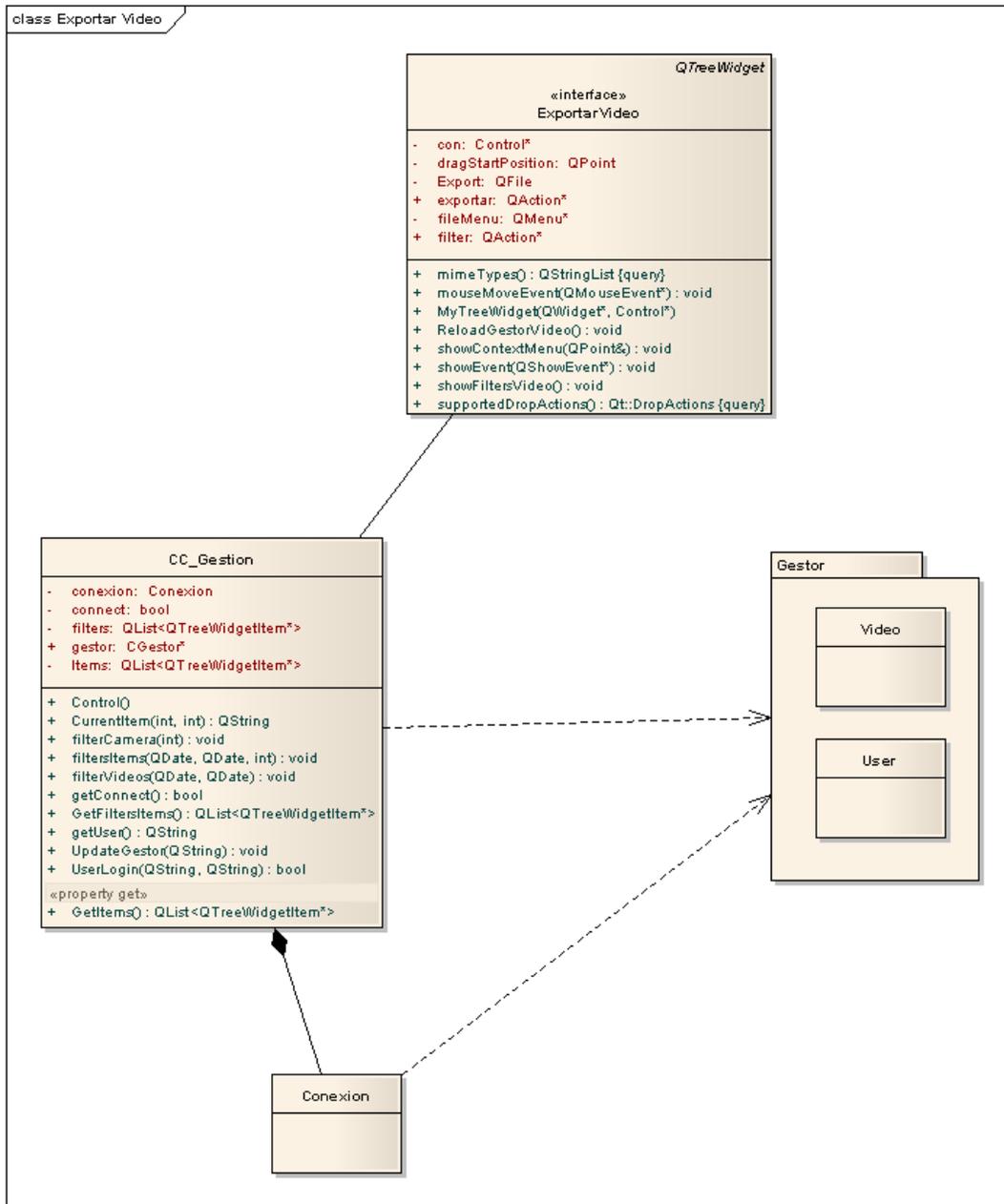


Figura 23. Diagrama de clases del diseño *CU_ExportarVideo*

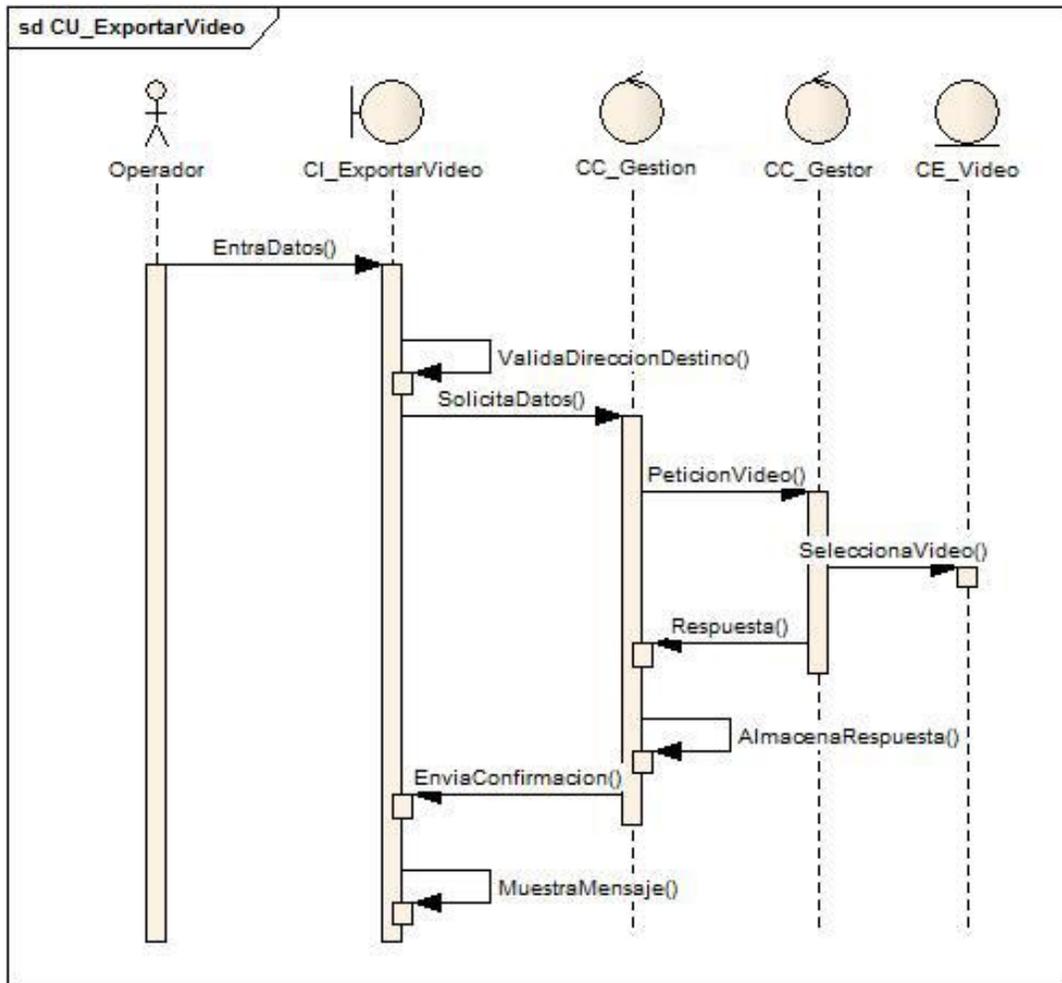


Figura 24. Diagrama de secuencia *CU_ExportarVideo*

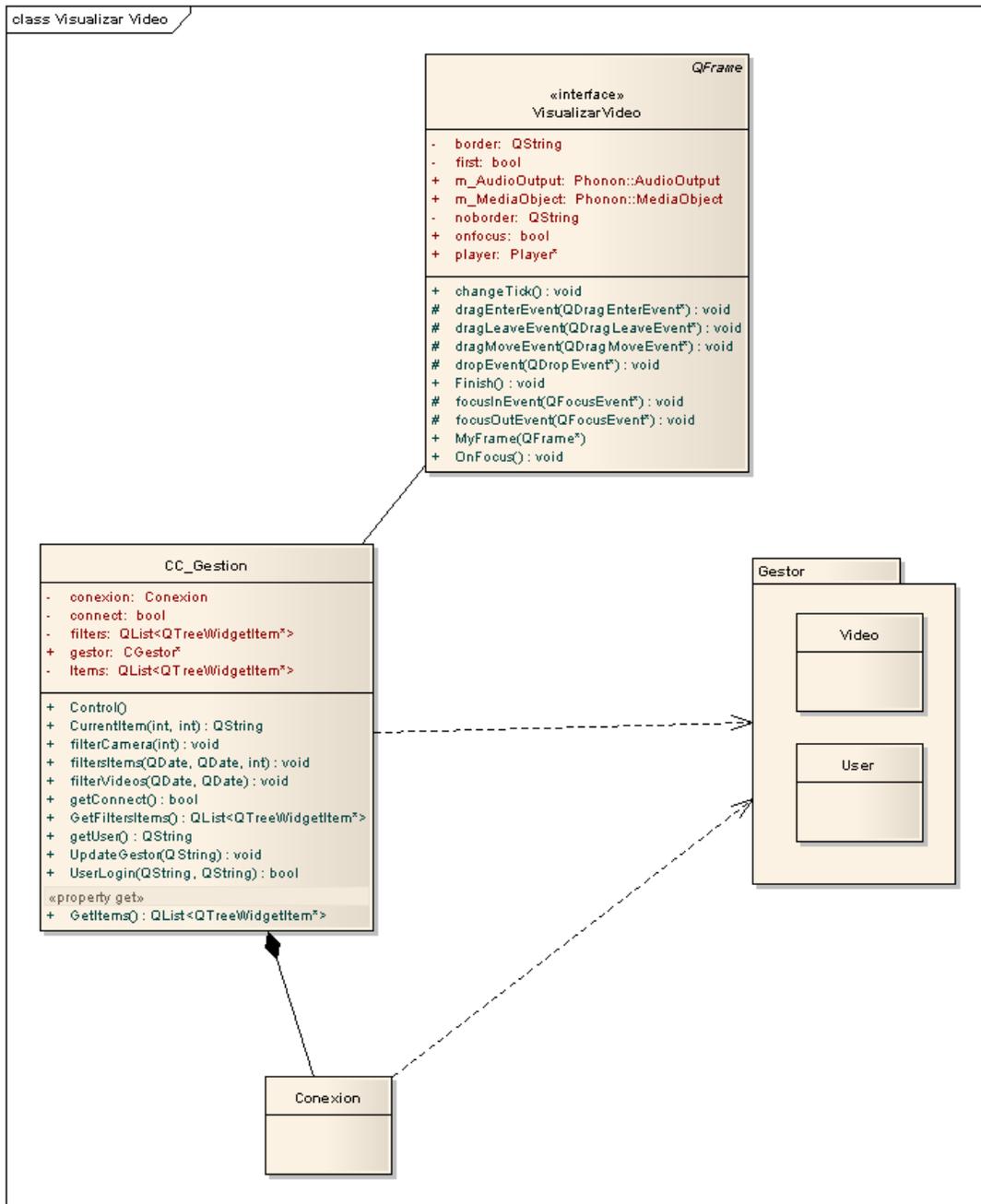


Figura 25. Diagrama de clases del diseño *CU_VisualizarVideo*

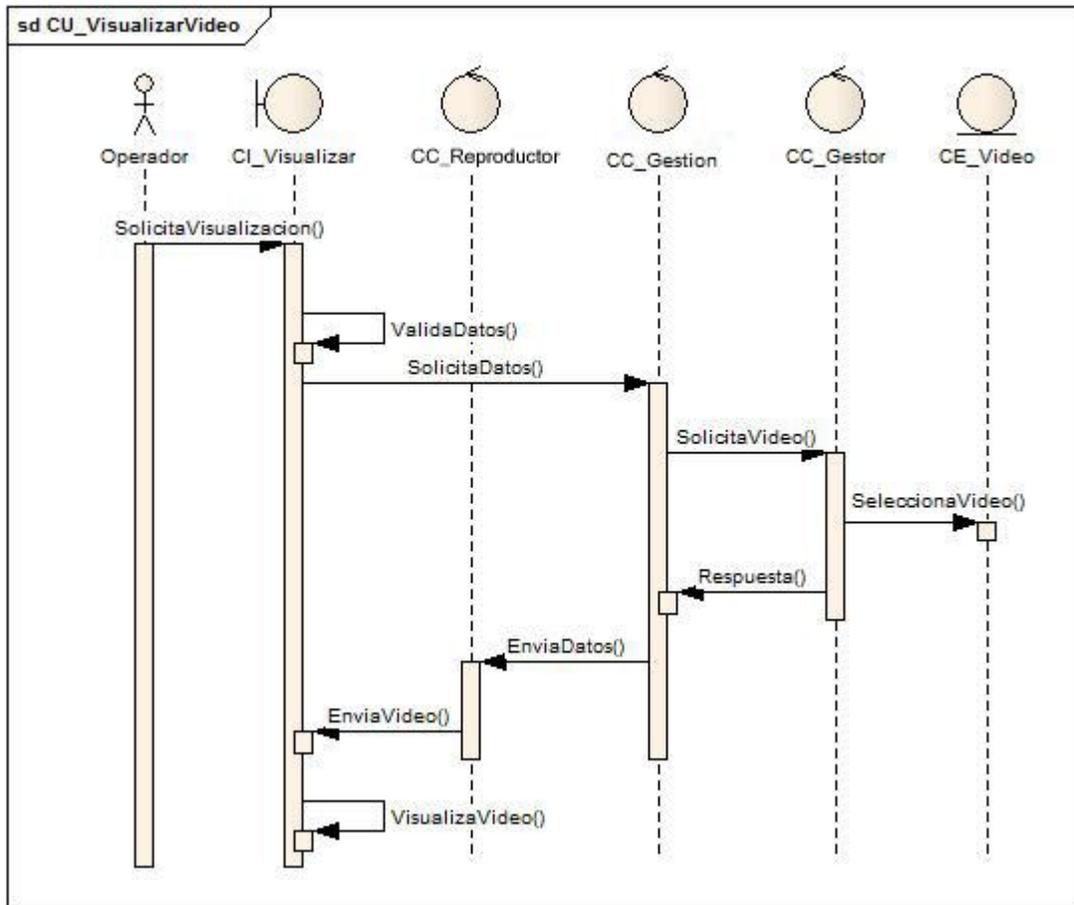


Figura 26. Diagrama de secuencia CU_VisualizarVideo

3.4. Descripción de las clases.

Nombre: Gestión	
Tipo de clase : Controladora	
Atributo	Tipo
ptr_gestor	CGestor*
items	QList<QTreeWidgetItem*>
filters	QList<QTreeWidgetItem*>
Para cada responsabilidad:	

Nombre:	Control()
Descripción:	Constructor que inicializa el manager con los datos requeridos.
Nombre:	GetItems()
Descripción:	Devuelve una lista con todos los Items (dígase cámaras y sus videos asociados) encontrados en el gestor.
Nombre:	UpdateGestor(Qstring dir)
Descripción:	Actualiza la dirección del gestor.
Nombre:	CurrentItem(int current,int parent)
Descripción:	Devuelve la dirección del video seleccionado.
Nombre:	FiltersItems(QDate DateStart,QDate DateEnd,int Cam)
Descripción:	Filtra los videos según criterios de búsqueda.
Nombre:	GetFiltersItems()
Descripción:	Devuelve los elementos filtrados.
Nombre:	UserLogin(QString user,QString pass)
Descripción:	Permite al usuario entrar al sistema.

Nombre: MainWindow	
Tipo de clase : Interfaz	
Atributo	Tipo
frame	FourView*
menu	MenuBar*
fileMenu	QMenu*
worker	Thread*
FrameFocus	MyFrame*

controlClass	Controller*
treecamera	MyTreeWidget*
fil	Filter*
Para cada responsabilidad:	
Nombre:	Principal(QWidget* parent=0)
Descripción:	Constructor que inicializa el reproductor con los datos requeridos.
Nombre:	getOnfocusFrame()
Descripción:	Toma el Reproductor que esta seleccionado.
Nombre:	enterVideoFullScreen(QString name,bool enable)
Descripción:	Muestra el reproductor a pantalla completa.
Nombre:	Function(char type)
Descripción:	Se encarga de las funciones del reproductor como play, pause y otras.
Nombre:	UpdateBars()
Descripción:	Actualiza las barras de reproducción de volumen y las de propiedades del video.
Nombre:	UpdateTime()
Descripción:	Actualiza el tiempo de reproducción del video.
Nombre:	stateChanged(Phonon::State newstate,Phonon::State oldstate)
Descripción:	Permite capturar cualquier cambio de estado ocurrente en el reproductor.
Nombre:	FourStyle()
Descripción:	Permite modificar la vista de reproducción a 4 reproductores.
Nombre:	Exporting()
Descripción:	Comienza la exportación del video seleccionado en un hilo "worker".
Nombre:	EndExport()
Descripción:	Finaliza la exportación del video.

Nombre:	UpdateGestor()
Descripción:	Muestra una interfaz que permite actualizar los datos del gestor.
Nombre:	FilterVideos()
Descripción:	Muestra una interfaz que permite filtrar los videos almacenados.

3.5. Conclusiones del capítulo.

En este capítulo primeramente se elaboraron los diagramas de clases del análisis donde se exponen los conceptos básicos del sistema, sus partes y relaciones. Luego los diagramas de clases del diseño y los diagramas de interacción por cada uno de los casos de usos, y por último las descripciones de las clases del diseño.

CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBA.

La implementación comienza con el resultado del diseño y se implementa en términos de componentes, es decir, ficheros de código fuente, ficheros de código binario, ejecutables y similares. El flujo de trabajo de implementación detalla cómo los elementos del modelo del diseño se implementan en términos de componentes y representa cómo se organizan en el modelo de despliegue, describiendo los componentes a construir, su organización y dependencias entre los nodos físicos en la que funcionará la aplicación.

4.1. Modelo de Implementación.

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema.

4.1.1. Diagrama de Despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física de un sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. En esta vista se realiza una representación gráfica de los nodos físicos en los que estará desplegado el sistema propuesto y la comunicación entre ellos. La aplicación desarrollada consta de 4 elementos fundamentales: PC Recuperador, Servidor de Grabación, PC Gestor y el Servidor de Base de Datos.

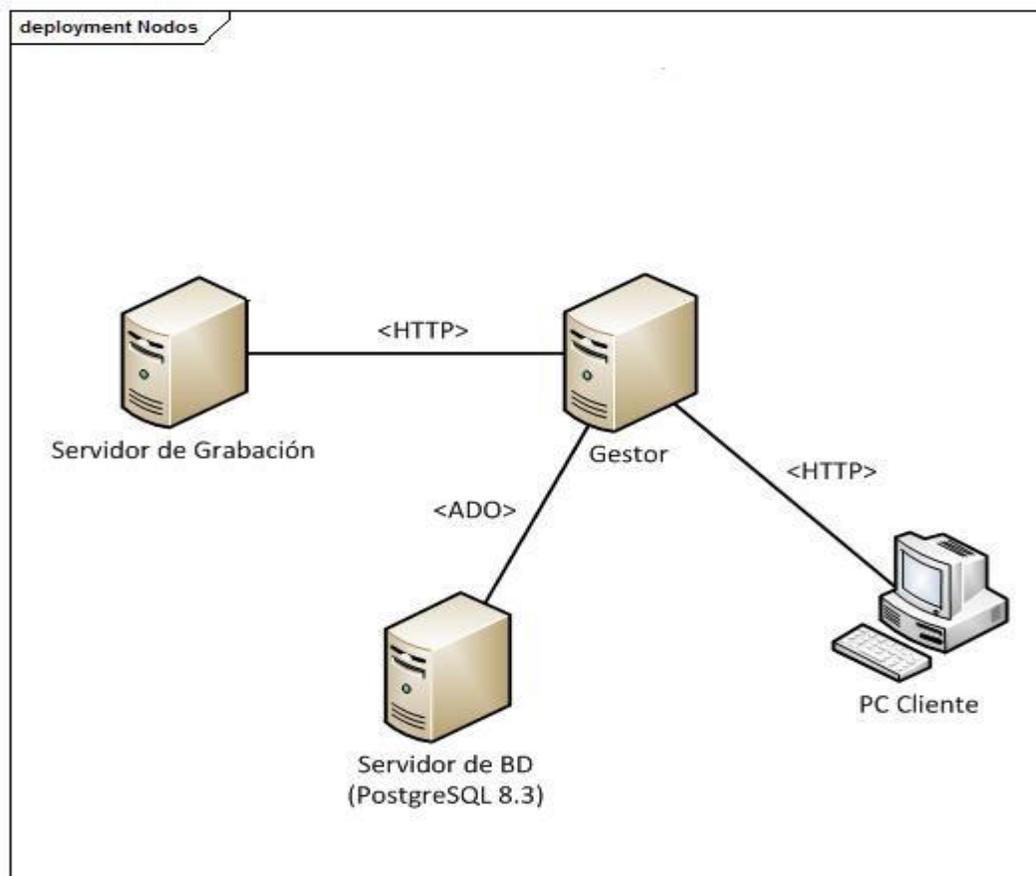


Figura 27. Diagrama de Despliegue

4.1.2. Diagrama de Componentes.

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos del sistema. El Diagrama de Componente muestra la relación entre los componentes del software, sus dependencias, comunicaciones, localización y otras condiciones.

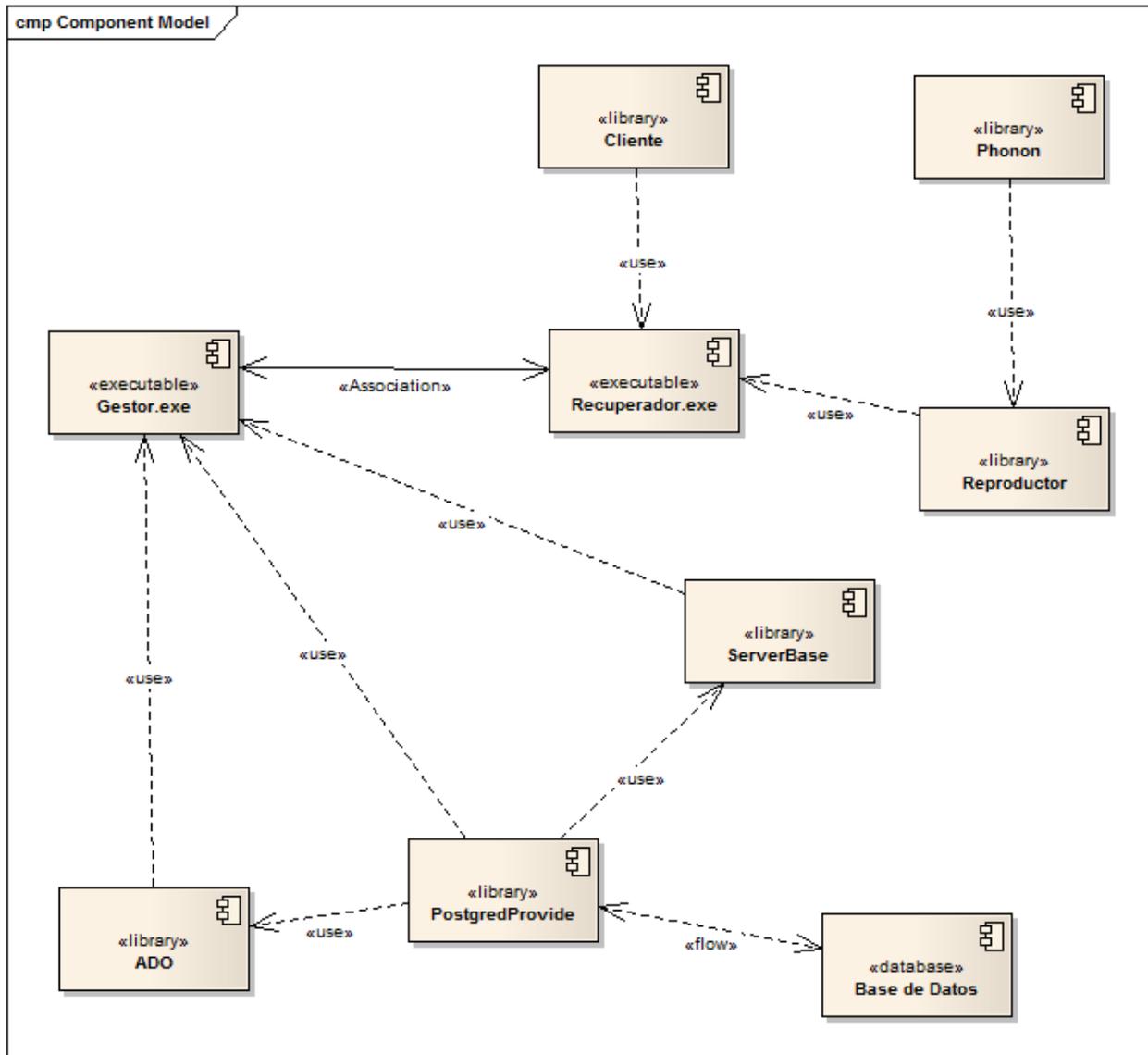


Figura 28. Diagrama de Componentes

4.2. Validación de la Solución Propuesta.

En el ciclo de vida de un software se realizan pruebas de software, con el fin de corregir errores que podrían comenzar desde el primer momento del proceso en que los objetivos estén planteados incorrectamente, así en sus últimos pasos, como diseño y desarrollo. Para garantizar la calidad en el desarrollo de un software primeramente se debe tener conocimiento de ¿Qué se entiende por pruebas de software?

4.2.1. Pruebas de Software.

Cuando se habla de las pruebas de software se puede definir de la siguiente manera: Proceso realizado concurrentemente a través de las diferentes etapas de desarrollo de software, cuyo objetivo es apoyar la disminución del riesgo de aparición de fallas y faltas en operación.(Cárdenas, 2009)

La prueba del software es un elemento crítico para la garantía de la calidad del Software. El objetivo de la etapa de pruebas es garantizar la calidad del producto desarrollado. Además, esta etapa implica:

- Verificar la interacción de componentes.
- Verificar la integración adecuada de los componentes.
- Verificar que todos los requisitos se han implementado correctamente.
- Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el software al cliente.
- Diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

La prueba es un proceso que se enfoca sobre la lógica interna del software y las funciones externas. La prueba es un proceso de ejecución de un programa con la intención de descubrir un error. Un buen caso de prueba es aquel que tiene alta probabilidad de mostrar un error no descubierto hasta entonces. Una prueba tiene éxito si descubre un error no detectado hasta entonces(Usaola, 2005)

Las pruebas de software definitivamente son imprescindibles para una mejor calidad en la terminación de un producto, para acercarse a un software sin errores se debe ante todo tener un conocimiento de las mismas y lograr escoger cuales deben aplicarse según se corresponda. Deben sostenerse en las diferentes etapas de desarrollo del software para controlar las deficiencias y detectarlas en el tiempo requerido, así luego corregir, ahorrando tiempo y costo.

4.2.2. Técnicas de Prueba.

Las técnicas de pruebas permitirán examinar los detalles procedimentales del código y la lógica del programa así como la interfaz y su funcionalidad. Por medio de las diferentes técnicas se brindan criterios variados para generar casos de pruebas que provoquen fallos en los programas. Es por ello que es de vital importancia conocer dichas técnicas.

Cualquier producto de ingeniería se puede probar de dos formas:

- Pruebas de caja negra: Realizar pruebas de forma que se compruebe que cada función es operativa.
- Pruebas de caja blanca: Desarrollar pruebas de forma que se asegure que la operación interna se ajusta a las especificaciones, y que todos los componentes internos se han probado de forma adecuada.

Pruebas de Caja Blanca

Las pruebas de Caja Blanca se basan en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que examinen que están correctas todas las condiciones y/o bucles para determinar si el estado real coincide con el esperado o afirmado. Esto genera gran cantidad de caminos posibles por lo que hay que dedicar esfuerzos a la determinación de las condiciones de prueba que se van a verificar.

Requieren del conocimiento de la estructura interna del programa y son derivadas a partir de las especificaciones internas del diseño o el código. Mediante los métodos de prueba de la Caja Blanca, el ingeniero de software puede obtener casos de prueba que garanticen que:

- Se ejerciten por lo menos una vez todos los caminos independientes para cada módulo.
- Se ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsa.
- Ejecuten todos los bucles en sus límites y con sus límites operacionales.
- Se ejerciten las estructuras internas de datos para asegurar su validez.

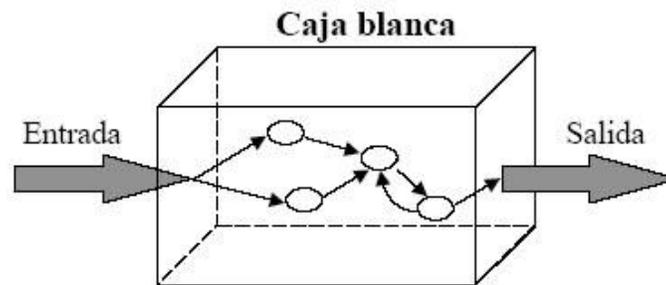


Figura 29. Enfoque de diseño de pruebas de caja blanca

Prueba de Caja Negra

El método de Caja Negra se enfoca en los requisitos fundamentales del software y permite obtener entradas que prueben todos los requisitos funcionales del programa. No considera la codificación dentro de sus parámetros a evaluar, es decir, que no están basadas en el conocimiento del diseño interno del programa. Con este equipo de pruebas se intenta encontrar:

- *Funciones incorrectas o ausentes.*
- *Errores de interfaz.*
- *Errores en estructuras de datos o en accesos a la bases de datos externas*
- *Errores de rendimiento.*
- *Errores de inicialización y terminación.*

Los casos de prueba de la Caja Negra pretende demostrar que:

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

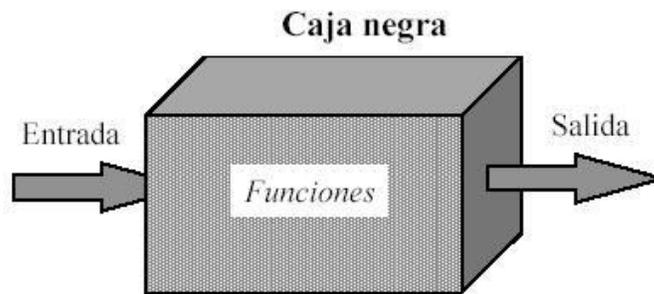


Figura 30. Enfoque de diseño de pruebas de caja negra

4.2.3. Diseño de los casos de pruebas (DCP) Caja Negra. Partición equivalente.

Nombre del Caso de Uso: Autenticar Usuario.

Descripción General: El caso de uso se inicia cuando se introduce el usuario y la contraseña para acceder en la aplicación, estos son verificados; el caso de uso finaliza mostrando la interfaz correspondiente.

Condiciones de Ejecución: El usuario debe existir en la Base de Datos y estar habilitado en el sistema.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Autenticar usuario	EC 1.1: Autenticar usuario satisfactoriamente	<p>El actor escribe su nombre de usuario y contraseña y presiona el botón “Aceptar” para entrar en el sistema.</p> <p>El sistema verifica que el usuario exista y la validez de la contraseña introducida. Verifica los permisos que tiene asignados y en dependencia de los permisos le permite acceder a la aplicación.</p>

EC 1.2: Autenticar usuario falla.	Si los datos de autenticación introducidos por el actor no son válidos, el sistema muestra el siguiente mensaje: "Usuario o contraseña incorrecta". El sistema da la posibilidad de volverse a introducir los datos.
-----------------------------------	--

Tabla 7. SC 1: Autenticas Usuario

Nombre del Caso de Uso:Exportar videos.

Descripción General:El caso de uso se inicia cuando se introduce la dirección donde se desea salvar el video, esta es verificada; el caso de uso finaliza mostrando un mensaje de aviso correspondiente.

Condiciones de Ejecución:La dirección debe ser válida y poseer permisos de escritura.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 2: Exportar videos	EC 2.1: Exportar video satisfactoriamente.	<p>El actor escoge la dirección donde desea guardar el video, presiona el botón “Aceptar” para comenzar a salvar el video.</p> <p>El sistema verifica que la dirección introducida sea válida. Verifica los permisos que tiene asignados y en dependencia de los permisos le permite exportar los videos.</p>

EC 2.2: Exportar videosfalla.	Si dirección no es válida o no posee permisos de escritura, el sistema muestra el siguiente mensaje: "Exporting Error". El sistema da la posibilidad de volverse a introducir los datos.
-------------------------------	--

Tabla 8. SC 2: Exportar videos

Nombre del Caso de Uso: Buscar videos.

Descripción General: El caso de uso se inicia cuando se introducen las fechas de inicio y fin de grabación del video o la cámara por la que fue grabada la secuencia de datos, estos son verificados; el caso de uso finaliza mostrando la interfaz con los videos correspondientes.

Condiciones de Ejecución: Las fechas de inicio y fin de grabación deben ser válidas.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 3: Buscar videos	EC 3.1: Buscar videos satisfactoriamente	<p>El actor selecciona o escribe la fecha de grabación del video que desea buscar y la cámara por la que fue grabado y presiona el botón “Aceptar” para comenzar búsqueda.</p> <p>El sistema verifica que las fechas de inicio y fin de la búsqueda sean válidas y en dependencia de eso muestra una interfaz con los videos relacionados con la búsqueda.</p>

<p>EC 3.2: Autenticar usuario falla.</p>	<p>Si las fechas de búsqueda introducidas por el autor no son válidos, el sistema muestra el siguiente mensaje: "Date Error". El sistema da la posibilidad de volverse a introducir los datos.</p>
--	--

Tabla 9. SC 3: Buscar videos

4.2.4. Resultados de las pruebas aplicadas

Sección	Resultado esperado	Resultado de la prueba
Autenticar usuario	Que se muestre la interfaz correspondiente según los permisos que posee el usuario.	Autentica al usuario correctamente y muestra la interfaz correspondiente sus permisos.

Exportar videos	Que el video sea exportado satisfactoriamente.	Exporta el video utilizando la dirección correspondiente.
Buscar videos	Que se muestren los videos correspondientes según filtros introducidos por el usuario.	Busca los videos según los filtros y los muestra en una interfaz.

Tabla 10. Resultados de las pruebas aplicadas.

4.3. Conclusiones.

En este capítulo se da una visión general de cómo quedó finalmente implementada la aplicación. Se muestra el modelo de despliegue y el diagrama de componentes del sistema propuesto, permitiendo un mejor entendimiento de la distribución física y lógica del sistema. Se realizaron tres pruebas de caja negra con el objetivo de detectar errores en el sistema.

CONCLUSIONES GENERALES

Con la realización de este trabajo se adquirieron conocimientos necesarios para el desarrollo del sistema y se logró cumplir con los objetivos y las tareas trazadas. Lo que permitió arribar a las siguientes conclusiones:

- Se realizó la caracterización de los principales sistemas de recuperación de videos digitales existente en la actualidad. Las principales limitaciones estuvieron en que estos softwares son en su mayoría privativos y con gran dependencia de sus fabricantes.
- Se describieron las principales tecnologías y herramientas actuales lo que permitió seleccionar las adecuadas para el desarrollo del sistema sentando las bases para que el producto final tenga la calidad requerida.
- Se implementó un sistema que brinda los servicios de búsqueda y visualización de todos los videos grabados por el sistema de video vigilancia. Además de gestionar toda la configuración referentes a las medias guardadas, cumpliendo con las funcionalidades trazadas.
- Se obtuvo una aplicación desktop, desarrollada en entornos libres y que da respuesta a la necesidad que la originó.

Es importante destacar la capacidad del sistema de ser ejecutado sobre diferentes sistemas operativo dígase Linux, Windows, Mac que es una de las limitaciones actuales de la mayoría de los sistemas en el mercado. Además todos los datos que usa o genera se acogen a los estándares internacionales que rigen estos sistemas, tanto para la compresión de imágenes, como para la trasmisión de las mismas.

RECOMENDACIONES

Durante el desarrollo del trabajo han surgido ideas que podrían implementarse en un futuro, de forma que se logren desarrollar más funcionalidades para la aplicación, para lo cual se recomienda:

- La implementación de un módulo de detección de movimientos.
- La implementación de un módulo para la detección de pérdida de foco.
- Incorporar una función de búsqueda inteligente que sea capaz de detectar eventos importantes en una Zona de interés señalada en el video.

GLOSARIO DE TÉRMINOS

Términos	Definiciones
IP	Protocolo de Internet (Internet Protocol), es un protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de paquetes conmutados no fiable de mejor entrega posible sin garantías.
CASE	Ingeniería de Software Asistida por Ordenador (Computer Aided Software Engineering), Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.
UML	Lenguaje Unificado de Modelado (Unified Modeling Language), Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.
IDE	Entorno Integrado de Desarrollo (Integrated Develop Environment), es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.
RUP	Proceso Unificado Racional (Rational Unified Process), proceso de desarrollo de software, es la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

REFERENCIA BIBLIOGRÁFICA

- Axis Communications. (2008). *Coste total de propiedad(TCO) Comparativa entre los sistemas de vigilancia con cámaras IP y cámaras analógicas*. Axis Communications.
- MPEG LA®. (2004). *MPEG-4 Visual*. (MPEG LA) Recuperado el 20 de Noviembre de 2008, de MPEG LA: <http://www.mpegla.com/m4v/m4v-licensees.cfm>
- aura. (2010). *II JOINEA - 2010*. Recuperado el 2011, de <http://joinea.exactas-unam.dyndns.org/charlas/215.pdf>
- Axis. (s.f.). Recuperado el 2011, de <http://www.axis.com/es/>
- Axis Communications. (21 de 11 de 2008). *Axis Communications*. Recuperado el 21 de 11 de 2008, de Axis Communications: <http://www.axis.com>
- Cárdenas, M. C. (2009). *Pruebas de software*. Obtenido de Documentos de investigación. [Online]: www.acis.org.co/index.php?id=972 .
- Curso C++. (s.f.). Recuperado el 2011, de <http://usuarios.multimania.es/absurdosoyyo/tutc.html>
- ENTERPRISE ARCHITECT. (2007). Recuperado el 2011, de www.ejie.net/documentos/Herramientas/EA.%20Manual%20de%20instalacion%20en%20cliente%20v1.0.doc
- Herranz, A. (24 de July de 2008). *Video Vigilancia IP*. Recuperado el 23 de November de 2008, de PCWORLD PROFESIONAL: http://www.idg.es/pcworldtech/Video_vigilancia_IP:_un_Gran_Hermano_en_nuestra_re/art191729-comunicaciones.htm
- Información general de .NET Framework Remoting*. (s.f.). Recuperado el 25 de November de 2008, de Visual Studio: [http://msdn.microsoft.com/es-es/library/kwdt6w2k\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/kwdt6w2k(VS.80).aspx)
- Las 10 ventajas principales de Microsoft Office Visio 2007*. (s.f.). Recuperado el 25 de November de 2008, de Microsoft Office Online: <http://office.microsoft.com/es-es/visio/HA101650313082.aspx>
- Microsoft Visual Studio - Wikipedia, la enciclopedia libre*. (s.f.). Recuperado el 25 de November de 2008, de Wikipedia, la enciclopedia libre: http://es.wikipedia.org/wiki/Microsoft_Visual_Studio#Visual_Studio_2008
- Panasonic. (25 de April de 2007). *Construnario.com - Notiweb: Sistemas de Seguridad Panasonic: 50 años haciendo la vida más fácil*. Recuperado el 25 de November de 2008, de Construnario >> Notiweb: http://www.construnario.es/notiweb/titulares_resultado.asp?regi=15902

- PostgreSQL8.3 ya disponible.* (s.f.). Recuperado el 25 de November de 2008, de Portal de Astra NTi:
<http://www.astra.es/noticias/postgresql-8-3-ya-disponible>
- Scati Labs - Wikipedia, la enciclopedia libre.* (s.f.). Recuperado el 25 de November de 2008, de Wikipedia, la enciclopedia libre: http://es.wikipedia.org/wiki/Scati_Labs
- Sony: Vigilancia inteligente en Verbania: España.* (s.f.). Recuperado el 25 de November de 2008, de Sony: Soluciones profesionales, corporativas y de broadcast: España:
http://www.sony.es/biz/view/ShowContent.action?site=biz_es_ES&contentId=1189437949251§iontype=NVM+CaseStudies
- synergix. (10 de 07 de 2008). *Tecnología y Synergix.* Recuperado el 16 de 02 de 2011, de Tecnología y Synergix: <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>
- Usaola, D. M. (2005). *Mantenimiento Avanzado de Sistemas de Información. Pruebas del Software.* Ciudad Real.
- Videovigilancia.* (s.f.). Recuperado el 30 de November de 2008, de TELENORMA Solucines en comunicación:
http://www.telenorma.com.co/index.php?option=com_content&view=section&layout=blog&id=3&Itemid=7
- Visual SourceSafe 2005.* (s.f.). Recuperado el 25 de November de 2008, de Visual Studio:
<http://msdn.microsoft.com/es-es/vcsharp/aa718670.aspx>
- vivotek. (s.f.). Recuperado el 2011, de
<http://www.vivotek.com/products/model.php?soft=ST7501&newlang=es&wish=ST7501>
- Wikipedia. (19 de Noviembre de 2008). *Proceso Unificado de Rational.* Recuperado el 29 de Noviembre de 2008, de Wikipedia: <http://es.wikipedia.org/wiki/RUP>
- Wikipedia. (22 de Noviembre de 2008). *Programación Extrema.* Recuperado el 29 de Noviembre de 2008, de Wikipedia La Enciclopedia Libre:
http://es.wikipedia.org/wiki/Programaci%C3%B3n_Extrema