

Universidad de las Ciencias Informáticas

Facultad 6



Sistema integral de vigilancia Suria: Análisis del módulo Grabador v2.0

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

Autor: Lidianis Aguila Valdés

Tutor: Ing. Magdiel Rivero González

Ciudad de La Habana, 21 de Junio de 2011

“Año 53 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los 21 días del mes de Junio del año 2011.

Autor:

Lidianis Aguila Valdés

Tutor:

Ing. Magdiel Rivero González

DATOS DE CONTACTO

Tutor: Ing. Magdiel Rivero González

Graduado de Ingeniero en Ciencias Informáticas en la UCI en el año 2010. Trabajador del Centro GE YSED en el Proyecto Sistema Integral de Vigilancia Suria en el cual se desempeña como Analista y Desarrollador.

Correo electrónico: mriverog@uci.cu

DEDICATORIA

Dedico este trabajo:

A mi mamá Mercedes y a mi papá Luis Ramón por dar todo de sí para que cada día yo sea una persona mejor.

A mi hermano Amaury y a mi cuñada Mayelín por apoyarme siempre y permitirme ser su guisazo.

A mis sobrinas Nileyam y Nayelís para que les sirva de ejemplo y lleguen a ser alguien en la vida con sueños y aspiraciones igual que su tía.

Al amor de mi vida, mi novio Osmel, que no por ser el último es menos importante para mí. Este trabajo es para ti mi amor por estar siempre ahí y además de ser mi novio, ser mi amigo, mi compañero y mi guía en todo momento. Te amo mucho y para siempre.

AGRADECIMIENTOS

Quiero agradecer:

A mis padres por quererme tanto y ser las personas más importantes de mi vida. Por apoyarme siempre y respetar mis decisiones. Por todo su esfuerzo y dedicación, los quiero mucho.

A mi hermano por estar siempre conmigo. Por ayudarme, aconsejarme y por estar siempre conmigo.

A mi cuñada por ser mi hermana, mi amiga, mi cómplice y por quererme como me quiere. Quiero que sepas que siempre vamos a estar juntas.

A mi novio por todo su amor y apoyo en estos años. Quiero que sepas que ni todos los agradecimientos, ni todas las dedicatorias, ni todas las palabras del mundo van a alcanzar para decirte lo mucho que te amo y que siempre estas presente en cada momento de mi vida.

A mis sobrinas por ser mis niñas lindas y la luz de mis ojos.

Al resto de mi familia que no nombro porque son muchos y no quisiera que se me quedara nadie, pero los quiero mucho y les agradezco todo el cariño y apoyo que me han dado siempre.

A mi tutor Magdiel por su ayuda y el apoyo que me ha dado durante el desarrollo del trabajo.

A la Universidad de las Ciencias Informáticas por darme la oportunidad de superarme en la especialidad que escogí.

RESUMEN

Los sistemas de video vigilancia han tenido una evolución acelerada impulsada por los avances tecnológicos, disminuyendo así la preocupación existente por la seguridad pública y privada. Además se han disminuido los precios de todos los equipos necesarios para su montaje, haciendo que la instalación de cámaras de captura de video no represente grandes inversiones económicas. Aunque en muchos países los sistemas de video vigilancia son muy comunes, a nuestro país debido a su difícil situación económica se le hace más trabajoso el acceso a los mismos. Debido a esto en la Universidad de las Ciencias Informáticas se está desarrollando un sistema de video vigilancia (Suria SIV) el cual brinda funcionalidades que también satisfacen las necesidades de las empresas e instituciones cubanas.

Con esta investigación, se pretende obtener toda la documentación necesaria para dar paso a la creación del diseño de una nueva versión del módulo Grabador encargado de la grabación de video a través del análisis del mismo. Además se espera lograr un mejor entendimiento del funcionamiento del sistema por parte de los clientes que harán uso de este.

Para lograr esto fue necesario determinar la metodología de desarrollo y la herramienta CASE (Ingeniería de Software Asistida por Computadora) que permitieron desarrollar el proceso ingenieril del sistema. Una vez terminada la solución propuesta se hizo preciso aplicar métricas para la validación de requisitos que permitieron eliminar ambigüedades que en un futuro pudieran provocar errores.

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	5
1.1. INTRODUCCIÓN.....	5
1.2. CONCEPTOS PRINCIPALES	5
1.3. EVOLUCIÓN DE LOS SISTEMAS DE GRABACIÓN DE VIDEO	5
1.4. PRINCIPALES CATEGORÍAS DE LOS VGD (VIDEO GRABADORES DIGITALES)	7
1.5. EMPRESAS PRODUCTORAS DE SISTEMAS DE GRABACIÓN DE VIDEO A NIVEL INTERNACIONAL	8
1.6. EMPRESAS PRODUCTORAS DE SISTEMAS DE GRABACIÓN DE VIDEO A NIVEL NACIONAL	10
1.7. PROCESO DE GRABACIÓN DE VIDEO EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS	10
1.8. EL LENGUAJE UNIFICADO DE MODELADO (UML) COMO SOPORTE DE LA MODELACIÓN DE LA SOLUCIÓN PROPUESTA.....	11
1.9. METODOLOGÍAS DE DESARROLLO.....	13
1.10. ¿POR QUÉ RUP?	19
1.11. HERRAMIENTAS CASE	19
1.12. ¿POR QUÉ ENTERPRISE ARCHITECT?.....	21
1.13. CONCLUSIONES PARCIALES	21
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA	22
2.1. INTRODUCCIÓN.....	22
2.2. MODELO DE DOMINIO.....	22
2.3. REQUERIMIENTOS FUNCIONALES DEL SISTEMA	23
2.4. REQUERIMIENTOS NO FUNCIONALES DEL SISTEMA	25
2.5. DEFINICIÓN DE LOS ACTORES Y CASOS DE USO DEL SISTEMA	28
2.6. PATRONES PARA LA IDENTIFICACIÓN DE LOS CASOS DE USO DEL SISTEMA	35
2.7. DIAGRAMAS DE CASOS DE USO DEL SISTEMA	37
2.8. MODELO DE ANÁLISIS	40
2.9. CONCLUSIONES PARCIALES.....	46
CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA	47
3.1. INTRODUCCIÓN.....	47
3.2. MÉTODO DE VALIDACIÓN DE LOS RESULTADOS	47

3.3.	APLICACIÓN DE LA MÉTRICA DE LA CALIDAD DE ESPECIFICACIÓN DE LOS REQUISITOS	47
3.4.	MATRIZ DE TRAZABILIDAD	51
3.5.	LISTA DE CHEQUEO	53
3.6.	MÉTRICAS DE CASOS DE USO DEL SISTEMA.....	55
3.7.	CONCLUSIONES PARCIALES	56
CONCLUSIONES GENERALES		57
RECOMENDACIONES.....		58
REFERENCIAS BIBLIOGRÁFICAS		59
BIBLIOGRAFÍA		61

TABLA DE IMÁGENES

FIGURA 1: MODELO DE DOMINIO	23
FIGURA 2: CONCORDANCIA (REUSABILIDAD)	36
FIGURA 3: CONCORDANCIA (ADICIÓN)	36
FIGURA 4: CRUD (COMPLETO).....	37
FIGURA 5: DIAGRAMA DE CASOS DE USO DEL SERVIDOR DE GRABACIÓN	37
FIGURA 6: DIAGRAMA DE CASOS DE USO DEL CLIENTE	38
FIGURA 7: DIAGRAMA DE CASOS DE USO DEL EDITOR DE CONFIGURACIÓN	39
FIGURA 8: DIAGRAMA DE CASOS DE USO DEL PLUGIN DE GRABACIÓN	39
FIGURA 9: CU-GRABAR	40
FIGURA 10: CU-MONITORIZAR	40
FIGURA 11: CU-CONFIGURAR	41
FIGURA 12: CU-BRINDAR INFORMACIÓN	41
FIGURA 13: CU-AUTENTICAR USUARIO.....	42
FIGURA 14: CU-GRABAR MANUALMENTE	42
FIGURA 15: CU-CONFIGURAR	43
FIGURA 16: CU-GESTIONAR CALENDARIO DE GRABACIÓN	43
FIGURA 17: CU-MOSTRAR CÁMARAS INSTALADAS	44
FIGURA 18: CU-VISUALIZAR CALENDARIO DE GRABACIÓN	44
FIGURA 19: ESCENARIO DETENER PROCESO DE GRABACIÓN	45
FIGURA 20: ESCENARIO INICIAR PROCESO DE GRABACIÓN	45
FIGURA 21: AUTENTICAR USUARIO	46

Sistema integral de vigilancia Suria: Análisis del **Introducción** módulo Grabador v2.0

INTRODUCCIÓN

A lo largo de los años el hombre ha tenido la necesidad de darle protección a sus bienes y para lograr esta seguridad se han invertido una gran cantidad de recursos. Poco a poco estas necesidades fueron incrementando y se fue despertando un gran interés de desarrollar formas de vigilancia más eficientes.

Debido a los avances tecnológicos que se han alcanzado, con el paso del tiempo fueron apareciendo los sistemas de video vigilancia los cuales fueron sustituyendo los antiguos métodos de vigilancia, además han logrado insertarse en una parte importante de la seguridad de grandes instituciones. Sus inicios se vieron marcados por los videos grabadores que contaban con el casete como medio de almacenamiento. Este permitía grabar un par de horas, lo cual traía como consecuencia, la necesidad de gran cantidad de cintas para dar respaldo al sistema, además de volver compleja la búsqueda de información específica en las mismas.

Los sistemas de vigilancia han evolucionado atravesando diferentes etapas en las últimas décadas. La primera generación de sistemas de vigilancia basada en video emplea señales y transmisión analógicas. Los sistemas analógicos emplean amplificadores de baja potencia y cables coaxiales para ver de forma remota las imágenes de las cámaras de seguridad. Las cámaras y los monitores están interconectados mediante una red de conmutación y distribución de video complejo y caro, que direcciona las imágenes de cada cámara hacia un monitor. Esta tecnología tiene demasiadas limitaciones y defectos. El video analógico solo se puede distribuir en una red local de cables coaxiales. La segunda generación de sistemas de vigilancia se basa, principalmente, en métodos de procesamiento y comunicación híbridos analógico-digitales. Aprovechan la flexibilidad ofrecida por los primeros algoritmos de procesado de video que permiten centrar la atención del operador humano en un grupo de situaciones de interés y, además, las facilidades proporcionadas por los primeros métodos de compresión digital para aprovechar el ancho de banda de transmisión. Debido a la demanda, estos sistemas fueron evolucionando hasta utilizar cámaras y servidores digitales para la grabación de video en un sistema completamente digitalizado. Actualmente existen sistemas modernos de tercera generación, basados en cámaras IP y redes de alta velocidad. Estos últimos aprovechan el progreso de las redes de ordenadores de bajo costo y alto rendimiento, y las comunicaciones multimedia fijas y móviles. Además constan de varias partes fundamentales: la dedicada a la visualización en tiempo real de los flujos de video, la de grabación de estos en dispositivos de almacenamiento, la que

Sistema integral de vigilancia Suria: Análisis del **Introducción** módulo Grabador v2.0

permite consultar los videos almacenados, así como una gran variedad de video sensores que utilizan el reconocimiento de patrones para brindar un producto de alto valor agregado. (1)

En la actualidad en Cuba se han realizado inversiones en varios equipos que son utilizados en las distintas esferas como son la científica, la educación y la salud. De esta última se puede decir que los centros de salud del país han recibido equipamientos que mejoran la calidad en los exámenes médicos o en dar sus resultados. Además existen hoteles, centros vacacionales y recreativos en los que se obtienen grandes beneficios monetarios los cuales ayudan al desarrollo económico del país así como a obtener éxito en el polo turístico que hoy existe. Para conservar todo esto se necesita de una seguridad estricta, pero los sistemas de video vigilancia a nivel mundial son muy costosos y al país se le hace más difícil el acceso a ellos. En el panorama nacional, se tiene una empresa (DATYS) que comercializa una solución de video vigilancia, pero al ser estos socios de Axis, su precio aumenta demasiado, siendo esta casi tan cara como las que brindan otras empresas a nivel internacional.

Para solucionar estos problemas en la Universidad de las Ciencias Informáticas se está desarrollando un sistema de video vigilancia. El mismo cuenta con varias funcionalidades como son la de visualización en tiempo real de los flujos de videos obtenidos de las cámaras y la que se dedica a recuperar los videos. También se encuentra la que le brinda funcionalidades de gestión de la información relacionada con las cámaras manejadas por el sistema y la grabación de flujos de video en dispositivos de almacenamiento.

La grabación y almacenamiento de los flujos de video es una de las principales funciones de los sistemas de video vigilancia. Ésta permite tener constancia de los hechos que ocurrieron en una fecha y hora determinada, lo cual constituye una prueba documental contundente ante la investigación de cualquier delito que se cometa. Por tal motivo se pretende obtener una versión más amplia del módulo de grabación que hoy existe lo cual brindaría una mayor eficiencia en el funcionamiento del sistema. Este nuevo Grabador debe brindar nuevas funcionalidades y ser multiplataforma.

La situación existente antes expuesta lleva a plantearse el siguiente problema a resolver **¿Cómo obtener la documentación ingenieril correspondiente al Análisis de una nueva versión del módulo de grabación del sistema de video vigilancia Suria que permita la creación de su diseño?** Tomando como objeto de estudio **el proceso de grabación de video digital desde las cámaras existentes en el sistema Suria** y delimitando como campo de acción **el Análisis de la nueva versión del módulo Grabador encargado del**

Sistema integral de vigilancia Suria: Análisis del **Introducción** módulo Grabador v2.0

proceso de grabación de los flujos de videos. Para dar solución al problema planteado se propone como objetivo general **Desarrollar el análisis del módulo Grabador v2.0 del sistema Suria, encargado del almacenamiento de los flujos de videos provenientes de las cámaras.**

Respondiendo al objetivo general se plantean las siguientes **tareas de la investigación:**

1. Realizar un estudio de los sistemas de video vigilancia que existen en la actualidad que permitan la grabación de múltiples flujos de videos en tiempo real.
2. Estudiar las metodologías y herramientas existentes que se usarán para la realización del análisis del sistema.
3. Realizar el Modelo de Dominio.
4. Realizar el Levantamiento de Requisitos.
5. Realizar el Modelo del Sistema.
6. Realizar el Modelo de Análisis.
7. Realizar la validación de la solución propuesta.

Se propone como **Idea a defender:** Con la documentación ingenieril correspondiente al análisis y una correcta validación de requisitos, se garantizan los artefactos necesarios para dar paso a la creación del diseño de la nueva versión del módulo de grabación del sistema de video vigilancia Suria SIV¹.

Los métodos utilizados en la investigación se explican a continuación:

Métodos Teóricos

Análisis histórico lógico: Se utiliza para hacer un estudio sobre la evolución y desarrollo que han tenido los sistemas de grabación de video y caracterizar el módulo de grabación existente en el sistema Suria.

Modelación: “La modelación es el método mediante el cual se crean abstracciones con el objetivo de explicar la realidad”. (2) Se utiliza para modelar el nuevo proceso de grabación del sistema Suria.

¹ SIV: Sistema Integral de Video Vigilancia

Sistema integral de vigilancia Suria: Análisis del **Introducción** módulo Grabador v2.0

Luego de darle cumplimiento al objetivo propuesto, se esperan como **posibles resultados**: Los artefactos ingenieriles obtenidos en el análisis de la nueva versión del módulo de grabación del sistema Suria.

- Modelo de Dominio.
- Requerimientos (Funcionales, No Funcionales).
- Modelo del Sistema.
- Modelo de Análisis.

El presente trabajo de diploma consta de tres capítulos:

Capítulo 1: Fundamentación teórica.

Capítulo 2: Características del sistema.

Capítulo 3: Validación de la solución propuesta.

A continuación se brinda una breve descripción acerca del contenido de cada capítulo:

Capítulo 1: Fundamentación teórica

Se mencionan algunos conceptos relacionados con el tema a tratar, se describen algunas de las características de los Sistemas de Grabación de Video Digital y se habla sobre la actualidad de los mismos en la Universidad de las Ciencias Informáticas. Además, se realiza un análisis de las metodologías y herramientas actuales, seleccionando las que serán utilizadas para la realización del trabajo.

Capítulo 2: Características del sistema

Se expresa mediante el modelo de dominio y se resaltan aquellas actividades que serán automatizadas. Se mencionan y especifican los requisitos funcionales y no funcionales que deben cumplir el sistema y se modela el mismo. Además se obtiene el diagrama de casos de uso del sistema, su descripción así como los diagramas de clases del análisis y diagramas de colaboración.

Capítulo 3: Validación de la solución propuesta

Se emplean distintas métricas que permitirán erradicar problemas en los requisitos o casos de uso del sistema, lo cual permite que se logren los objetivos propuestos satisfactoriamente.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En este capítulo se abordarán algunos temas relacionados con el dominio del problema, con el objetivo de facilitar una mayor comprensión del tema. Se explica detalladamente el funcionamiento, características y funcionalidades de sistemas existentes y se presenta el análisis realizado para seleccionar la metodología y la herramienta CASE que se utiliza en el desarrollo ingenieril de la nueva versión del módulo de grabación existente en el sistema Suria.

1.2. Conceptos principales

1.2.1. *Video digital*

El vídeo digital es un tipo de sistema de grabación de vídeo que funciona usando una representación digital de la señal de vídeo. Es el resultado de la combinación de secuencias de imágenes, audio, texto y movimiento que se obtienen por muestreo y cuantificación de la señal de video analógica. Está compuesto por fotogramas, que son cada una de las imágenes individuales de una secuencia o animación, por tomas, que son las secuencias de imágenes realizadas en un mismo plano y las escenas, que representan un conjunto de tomas relacionadas entre sí. (3)

1.2.2. *Sistemas de grabación digital*

Un sistema de grabación digital es un sistema inteligente de procesamiento digital de video para control y auditoria, que otorga una solución de avanzada tecnología para el campo de la seguridad. (4)

1.3. Evolución de los sistemas de grabación de video

Existen reportes periodísticos que ubican el uso del primer Sistema de Video Vigilancia en el año 1969 por la policía de Nueva York, basado en un simple Circuito Cerrado de Televisión (CCTV), desde entonces hasta la fecha, los sistemas de video vigilancia han tenido un gran desarrollo hasta llegar a los modernos sistemas digitales. (5)

Al mismo tiempo que los Sistemas de Video Vigilancia han ido evolucionando, las maneras de almacenar los videos obtenidos de las cámaras también han tenido un constante desarrollo. Estos avances han contribuido

en gran medida a popularizar la video vigilancia, ya que permite preservar las pruebas de un hecho delictivo usándose frecuentemente como evidencia ante los tribunales.

A partir de la década de los setenta se comenzaron a utilizar los Sistemas de Video Vigilancia en el sector privado y ya para la década de los ochenta, era común el uso de los grabadores de video analógico en cinta (VCR). (5) Estas tecnologías tenían varias desventajas, una de ellas era el poco tiempo de almacenamiento que tenía cada cinta, por lo que debían ser cambiadas periódicamente y obviamente esto debía hacerse de forma manual, por otra parte, solo permitía almacenar videos desde una sola cámara, por lo que era necesario utilizar una grabadora por cada una de ellas. (6)

Para solucionar estos problemas se crearon los Multiplexadores Digitales, los cuales podían grabar a uno o menos frames por segundo, reduciendo así el espacio en cinta, necesario para almacenar el video. Esto posibilitaba no tener que cambiar la cinta hasta dentro de cuatro o cinco días, sin embargo, esta tarea aún debía ser realizada de forma manual, además se perdían detalles de los incidentes grabados. Los Multiplexadores además podían tomar los videos de varias cámaras y juntarlos en uno solo, esto hacía que el tamaño de las imágenes fuera muy reducido, pudiendo ser incluso de 1/16 del tamaño de la pantalla; sin posibilidades de aumentarla, lo que dificultaba distinguir los detalles de la escena. (6)

A esto se unía un rápido decrecimiento en la calidad del video por el uso continuo de las cintas, provocando que hubiera que reponerlas cada cierto tiempo, además estos dispositivos mecánicos se deterioraban rápidamente lo que representaba un gran gasto en su mantenimiento o sustitución. (6)

Los sistemas de Grabación de Video Digital (DVR, Digital Video Recorder) surgen a principios de los noventa. Su aspecto externo era similar al de los grabadores de cinta que eran operados por botones o mediante control remoto. Desafortunadamente, debido a limitaciones tecnológicas los primeros DVR's no eran fiables y poseían limitadas funcionalidades ofreciendo poca calidad de video, por esto no eran lo suficientemente potentes para brindar soluciones que pudieran manejar una gran cantidad de cámaras. (6)

1.4. Principales categorías de los VGD (Video Grabadores Digitales)

Actualmente los VGD's se han dividido en tres categorías muy especializadas:

1.4.1. *Dedicadas, autónomas (Non- PC based)*

Unidad compacta con entradas de video analógico para cable coaxial, que físicamente no tiene parecido a un PC, pero que en su arquitectura electrónica está basada en un esquema de cómputo tradicional (fuente, motherboard, disco duro, puertos de comunicaciones, tarjeta de captura de video, dispositivos de entrada y salida). Usa un sistema operativo propietario o cualquier versión comercial embebida y un software dedicado a grabar video. Principalmente estas soluciones utilizan lo que se conoce como "ChipASIC" (Circuito Integrado de Aplicaciones Específicas, por sus siglas en inglés) estos chips están diseñados para un propósito específico por lo que no pueden ser reprogramados para agregarle nuevas funcionalidades. (7)

1.4.2. *Basadas en PC (PC based)*

Esta solución está basada en un equipo tipo servidor de misión crítica o incluso una PC casera, el cual tiene instalada una tarjeta capturadora de video de altas especificaciones y otros dispositivos como teclado y Joystick. Utiliza un sistema operativo que puede ser tradicional y un software dedicado a la grabación de video. Esta tecnología suele ser más flexible y adaptable, pues permite actualizar las técnicas de compresión y poseen una interfaz que hace más cómoda la interacción del usuario con la misma. Los últimos avances en este campo, son la utilización de los llamados Procesadores de Señales Digitales (DSP, de sus siglas en inglés) que como su nombre indica es un microprocesador especializado en realizar eficientemente cálculos sobre señales digitalizadas que eran originalmente analógicas. La principal ventaja de estos, es que son fácilmente programables. (7)

1.4.3. Grabadores de video basados en red (NVR²)

Estas soluciones parten de señales de video totalmente digitalizadas y emplean el Protocolo de Internet (IP). En este caso, la señal de video convertida en datos tradicionales de alta velocidad, es grabada en dispositivos digitales que reciben la información mediante una tarjeta de red (Tarjeta de Interfaz de Red, NIC, por sus siglas en inglés) de tipo Ethernet. Es decir, la videograbadora no tiene tarjetas captadoras de video, todo el tráfico de video entra y sale mediante la red de comunicaciones tipo LAN/WAN. Se emplean mucho, cuando se deben concentrar las señales de video provenientes de diversos puntos alejados y ya existe una red de datos que se pueda usar, con suficiente ancho de banda disponible para estos. En algunos casos se sugiere construir una red de datos totalmente separada de la tradicional, para no sacrificar desempeño. (7)

1.5. Empresas productoras de sistemas de grabación de video a nivel internacional

Existen varias empresas en el panorama internacional, que lideran el paso de tecnología analógica a digital en el mundo, entre ellas se encuentran:

Axis: Es una compañía de TI (Tecnologías de la Información) que ofrece soluciones de video IP dirigidas al mercado profesional. La compañía es líder del mercado del video IP, conduciendo el cambio de la video vigilancia analógica hacia las soluciones digitales. Axis es una empresa sueca que tiene oficinas en dieciocho países, y que coopera con socios comerciales en más de setenta países de todo el mundo. El nombre de su principal software de video vigilancia es *Axis Camera Station*, el cual funciona con cámaras de red y codificadores de vídeo de Axis y proporciona funciones de supervisión de vídeo, grabación y gestión de eventos. Los usuarios pueden realizar una grabación de vídeo continua, programada, activada por alarma y/o por detección de movimiento. El software dispone de múltiples funciones de búsqueda de eventos grabados, éste usa mpeg4 para el almacenamiento de los flujos, lo que es un inconveniente ya que es un formato propietario con licencias muy costosas. (8)

² Network Video Recorder

Panasonic: Se caracteriza por vender soluciones de seguridad completas más que hardware independiente (Axis lidera este mercado). Cada kit de solución es muy dependiente a un hardware específico. Panasonic ha hecho importantes innovaciones en el mundo del hardware de video vigilancia, este es el caso de la primera cámara con procesador de señal digital, con una tecnología actualmente conocida como i-Pro, pensada para la monitorización de imágenes on-line y la grabación en alta resolución. Además, su practicidad es máxima ya que su capacidad de comprensión de imágenes es dual, JPEG y MPEG4. Esta compañía comercializa soluciones de grabación de video que incluyen software y hardware, las cuales usan las más modernas tecnologías de compresión de video, así como el empleo de sistemas RAID y discos de estado sólido. También soportan un amplio rango de cámaras de otros fabricantes y son compatibles con Windows y Mac. Su principal deficiencia es que cada serie de cámaras viene con un software específico para ella, con poca o ninguna compatibilidad con otras cámaras, lo que dificulta mucho el trabajo para desarrolladores externos, porque tienen que soportar muchas APIs³ distintas”. (9)

SCATI LABS S.A: Es una empresa tecnológica radicada en Zaragoza, dedicada a la fabricación de equipos de grabación digital de videos y a la realización de consultorías de alto valor añadido, para el desarrollo de proyectos en los que se requiera una gestión de audio y video específica. Su principal producto es el *VisionSurfer* en varias versiones, el cual constituye una suite que incluye hardware y software integrados. El almacenamiento se realiza en un sistema rack o compacto, que ofrece altas velocidades de entrada de videos. Aunque sus soluciones son muy caras brindan la perfecta combinación de robustez, funcionalidad y servicio. (10)

Al valorar críticamente los productos de estas compañías, los autores concluyen que la mayoría son sistemas provistos por fabricantes de hardware, y tienen gran dependencia del mismo. La mayoría de las funcionalidades del sistema vienen implementadas en el hardware y el software solo se encarga de

³ Una API (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

gestionarlas. Lo que encarece el costo del sistema y dificulta su extensión con cámaras u otro equipo que no sea del fabricante que lo proveyó, ya que existe una muy baja compatibilidad.

1.6. Empresas productoras de sistemas de grabación de video a nivel nacional

DATYS, Tecnologías & Sistemas: Es una empresa que nace desde ACITED Informática en el año 2005 y adquiere personalidad jurídica propia el 16 de abril de 2007, para asumir el desarrollo de sistemas complejos y especializados de alta demanda en el mercado. Se distingue por desarrollar simultáneamente varias líneas de negocios agrupadas según diferentes objetivos, cuyo denominador común es el uso de tecnologías de avanzada y su acercamiento a estándares de calidad para productos existentes, de similar naturaleza, en el mercado internacional. Entre los productos desarrollados por esta empresa se encuentra *XYMA SaveVision*, el cual es un sistema de video vigilancia compatible con productos AXIS de video IP. (11)

Luego de haber analizado las características de esta empresa se puede decir que presenta dificultades similares a las de las compañías internacionales pues al ser esta socia de Axis sus productos solo son compatibles con hardware Axis. Esto es un inconveniente porque hace que sus soluciones sean muy costosas y que no se pueda utilizar su software con otro tipo de hardware.

1.7. Proceso de grabación de video en la Universidad de las Ciencias Informáticas

En la Universidad de las Ciencias Informáticas (UCI) existe hoy un sistema de video vigilancia llamado Suria el cual cuenta con distintos módulos como son el Gestor, Visor, Recuperador y uno de los más importantes es el Grabador que es el que se dedica a la grabación de video. Todo el funcionamiento del Grabador, así como su colaboración con otros módulos, es coordinado por el Gestor, el cual es el Repositorio activo dentro del sistema Suria.

El módulo de grabación está compuesto por varias aplicaciones: Servidor de Grabación, Plugins de Grabación, Cliente y Editor de Configuración.

El servidor de grabación: consistente en una aplicación tipo servicio de Windows, que atiende los pedidos de grabación apoyándose en un grupo de librerías denominadas plugins de grabación.

Un plugin de grabación: implementa la funcionalidad de grabación para uno o varios tipos de cámaras, usando una tecnología de grabación específica.

El cliente: es una aplicación gráfica que funciona como la interfaz de usuario principal del módulo, permitiendo la planificación de las grabaciones por calendario.

El editor de configuración: es una sencilla aplicación gráfica que permite modificar los parámetros de configuración del servidor de grabación.

El Grabador también cuenta con distintas funcionalidades para la grabación de videos, estas se pueden realizar de forma manual, mediante un grupo de reglas definidas para cada cámara o ante la ocurrencia de un determinado evento en el sistema las cuales funcionan con cámaras de diferentes fabricantes. Este módulo presenta varias dificultades que hacen que no sea del todo eficiente dentro de las que se encuentran: que no es multiplataforma y que puede brindar funcionalidades que no han sido implementadas. Por tal motivo el presente trabajo brinda la documentación necesaria para dar paso al desarrollo de un nuevo Grabador el cual corrija las dificultades que se presentan actualmente.

1.8. El Lenguaje Unificado de Modelado (UML) como soporte de la modelación de la solución propuesta

¿Qué es UML?

UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema. Este lenguaje nos indica cómo crear y leer los modelos. (12)

Funciones principales: (12)

- **Visualizar:** UML permite expresar de una forma gráfica un sistema de forma que otras personas lo puedan entender.
- **Especificar:** UML permite especificar cuáles son las características de un sistema antes de su construcción.

- **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.
- **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Un modelo UML está compuesto por tres clases de bloques de construcción: (12)

- **Elementos:** Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.).
- **Relaciones:** relacionan los elementos entre sí.
- **Diagramas:** Son colecciones de elementos con sus relaciones.

Diagramas UML: (12)

Un diagrama es la representación gráfica de un conjunto de elementos con sus relaciones. En concreto, un diagrama ofrece una vista del sistema a modelar.

Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas que permiten visualizar el sistema desde varias perspectivas.

- Diagrama de casos de uso.
- Diagrama de clases.
- Diagrama de objetos.
- Diagrama de secuencia.
- Diagrama de colaboración.
- Diagrama de estados.
- Diagrama de actividades.
- Diagrama de componentes.
- Diagrama de despliegue.

1.9. Metodologías de desarrollo

En los últimos años ha crecido notablemente el interés en la realización de productos de software, y para el desarrollo de estos existen dos tipos de metodologías: las ágiles y las tradicionales o robustas, dentro del gran grupo de metodologías se encuentran el Proceso Unificado de Desarrollo (RUP) y la Programación Extrema (XP), de las cuales se mencionarán algunas características para poder argumentar la que será seleccionada.

1.9.1. RUP

Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. (13)

Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). (13)

El RUP está basado en 5 principios claves que son: (13)

- Adaptar el proceso: El proceso deberá adaptarse a las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico. También se deberá tener en cuenta el alcance del proyecto.
- Balancear prioridades: Los requerimientos de los diversos participantes pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un balance que satisfaga los deseos de todos. Debido a este balanceo se podrán corregir desacuerdos que surjan en el futuro.
- Demostrar valor iterativamente: Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados.
- Elevar el nivel de abstracción: Este principio dominante, motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL o marcos de referencia (Frameworks) por nombrar

algunos. Esto evita que los ingenieros de software vayan directamente de los requisitos a la codificación de software a la medida del cliente, sin saber con certeza qué codificar para satisfacer de la mejor manera los requerimientos y sin comenzar desde un principio pensando en la reutilización del código. Un alto nivel de abstracción también permite discusiones sobre diversos niveles y soluciones arquitectónicas. Estas se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con el lenguaje UML.

- Enfocarse en la calidad: El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente.

Principales características: (13)

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

La metodología RUP divide en 4 fases el desarrollo del software: (13)

- **Inicio:** Se determina la visión del proyecto.
- **Elaboración:** Se determina la arquitectura óptima.
- **Construcción:** Se obtiene la capacidad operacional inicial.
- **Transmisión:** Se obtiene la liberación del proyecto.

Los elementos de RUP son: (13)

- **Actividades:** Son los procesos que se llegan a determinar en cada iteración.
- **Trabajadores:** Son las personas involucradas en cada proceso.

- **Artefactos:** Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. (13)

1.9.2. *XP*

Es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, Extreme Programming Explained: Embrace Change (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que estos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto, es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. (13)

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software. (13)

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizadas para proyectos de corto plazo, corto equipo y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. (13)

Principios básicos de la programación extrema: (13)

- Simplicidad: La simplicidad es la base de la programación extrema. Se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento. Un diseño complejo del código junto a sucesivas modificaciones por parte de diferentes desarrolladores, hacen que la complejidad aumente exponencialmente. Para mantener la simplicidad es necesaria la refactorización del código, esta es la manera de mantener el código simple a medida que crece. También se aplica la simplicidad en la documentación, de esta manera el código debe comentarse en su justa medida, intentando que el

código esté autodocumentado. Para ello se deben elegir adecuadamente los nombres de las variables, métodos y clases. Los nombres largos no disminuyen la eficiencia del código ni el tiempo de desarrollo gracias a las herramientas de autocompletado y refactorización que existen actualmente. Aplicando la simplicidad junto con la autoría colectiva del código y la programación por parejas se asegura que mientras más grande se haga el proyecto, todo el equipo conocerá más y mejor el sistema completo.

- Comunicación: La comunicación se realiza de diferentes formas. Para los programadores el código comunica mejor mientras más simple sea. Si el código es complejo hay que esforzarse para hacerlo inteligible. El código autodocumentado es más fiable que los comentarios, ya que estos últimos pronto quedan desfasados con el código a medida que es modificado. Debe comentarse solo aquello que no va a variar, por ejemplo el objetivo de una clase o la funcionalidad de un método. Las pruebas unitarias son otra forma de comunicación ya que describen el diseño de las clases y los métodos al mostrar ejemplos concretos de cómo utilizar su funcionalidad. Los programadores se comunican constantemente gracias a la programación por parejas. La comunicación con el cliente es fluida ya que el cliente forma parte del equipo de desarrollo. El cliente decide qué características tienen prioridad y siempre debe estar disponible para solucionar dudas.
- Retroalimentación: Al estar el cliente integrado en el proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real. Al realizarse ciclos muy cortos tras los cuales se muestran resultados, se minimiza el tener que rehacer partes que no cumplen con los requisitos y ayuda a los programadores a centrarse en lo que es más importante. Considérense los problemas que derivan de tener ciclos muy largos. Meses de trabajo pueden tirarse por la borda debido a cambios en los criterios del cliente o malentendidos por parte del equipo de desarrollo. El código también es una fuente de retroalimentación gracias a las herramientas de desarrollo. Por ejemplo, las pruebas unitarias informan sobre el estado de salud del código. Ejecutar las pruebas unitarias frecuentemente permite descubrir fallos debidos a cambios recientes en el código.
- Coraje o Valentía: Los puntos anteriores parecen tener sentido común, entonces, ¿por qué coraje? Para los gerentes la programación en parejas puede ser difícil de aceptar, parece como si la productividad se fuese a reducir a la mitad ya que solo la mitad de los programadores está escribiendo código. Hay que ser valiente para confiar en que la programación por parejas beneficia la calidad del código sin repercutir negativamente en la productividad. La simplicidad es uno de los

principios más difíciles de adoptar. Se requiere coraje para implementar las características que el cliente quiere, ahora, sin caer en la tentación de optar por un enfoque más flexible que permita futuras modificaciones. No se debe emprender el desarrollo de grandes marcos de trabajo (Frameworks) mientras el cliente espera. En ese tiempo el cliente no recibe noticias sobre los avances del proyecto y el equipo de desarrollo no recibe retroalimentación para saber si va en la dirección correcta. La forma de construir marcos de trabajo es mediante la refactorización del código en sucesivas aproximaciones.

Las características fundamentales del método son: (13)

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias⁴ continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión⁵.
- Programación en parejas: frecuente integración del equipo de programación con el cliente o usuario.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo

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

5 Prueba 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.

hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

- La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Mientras más simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.

¿Qué es lo que propone XP?

Derechos del cliente: (13)

- Decidir que se implementa.
- Saber el estado real y el progreso del proyecto.
- Añadir, cambiar o quitar requerimientos en cualquier momento.
- Obtener lo máximo de cada semana de trabajo.
- Obtener un sistema funcionando cada 3 o 4 meses.

Derechos del desarrollador: (13)

- Decidir cómo se implementan los procesos.
- Crear el sistema con la mejor calidad posible.
- Pedir al cliente en cualquier momento aclaraciones de los requerimientos.
- Estimar el esfuerzo para implementar el sistema.
- Cambiar los requerimientos en base a nuevos descubrimientos.

Lo fundamental en este tipo de metodología es: (13)

- La comunicación, entre los usuarios y los desarrolladores
- La simplicidad, al desarrollar y codificar los módulos del sistema
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales
- El cliente o el usuario se convierte en miembro del equipo.

1.10. ¿Por qué RUP?

Se selecciona RUP por ser una metodología eficaz que se adapta a las características propias del software que se está desarrollando, a través de la cual se pueden eliminar los riesgos que podrían presentarse durante el desarrollo del software. Permite enfocarse en trabajar de forma organizada, donde se controla y documenta todo lo relacionado con el proyecto que se esté realizando. Esta metodología ayuda a construir un software de alta calidad, que sea desarrollado en el tiempo planificado, además satisface la necesidad de ser elaborado de una forma más rápida.

1.11. Herramientas CASE

1.11.1. *Visual paradigm*

Visual paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Presenta licencia gratuita y comercial. Es fácil de instalar y actualizar y compatible entre ediciones. (14)

Características principales: (14)

- Soporte de UML versión 2.1.
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- Ingeniería inversa - Código a modelo, Código a diagrama.

- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- Editor de figuras.

1.11.2. *Rational rose*

Rational rose es una herramienta de producción y comercialización establecidas por Rational Software Corporation (actualmente parte de IBM). Rose es un instrumento operativo conjunto que utiliza el Lenguaje Unificado (UML) como medio para facilitar la captura de dominio de la semántica, la arquitectura y el diseño. (14)

Sus características principales: (14)

- No es gratuito, se debe hacer un previo pago para poder adquirir el producto.
- La ingeniería de código (directa e inversa) es posible para ANSI C++, Visual C++, Visual Basic 6, Java, J2EE/EJB, CORBA, Ada 83, Ada 95, Bases de datos: DB2, Oracle, SQL 92, SQL Server, Sybase, Aplicaciones WEB.
- Posee limitantes que la hacen débil en comparación a otras herramientas como Visual Paradigm, estas debilidades radican en la dependencia de la plataforma Windows y la integración solo con herramientas que estén en el mismo grupo de software propietario.

1.11.3. *Enterprise architect 7.0*

Enterprise architect es una herramienta CASE⁶ que combina el poder de la última especificación UML 2.1 con alto rendimiento, interfaz intuitiva para traer modelado avanzado al escritorio y para el equipo completo de desarrollo e implementación. Con un gran conjunto de características y un valor

⁶ Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador.

sin igual económicamente. Puede equipar a un equipo entero, incluyendo analistas, evaluadores, administradores de proyectos, personal del control de calidad, equipo de desarrollo y más, por una fracción del costo de algunos productos competitivos. Es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad, aspecto de vital importancia. Se encarga de administrar la complejidad con herramientas para rastrear las dependencias, brinda soporte para modelos muy grandes. Provee una generación poderosa de documentos y herramientas de reporte con un editor de plantilla completo. (15)

1.12. ¿Por qué Enterprise architect?

Se seleccionó Enterprise architect por ser esta una herramienta que soporta ocho de los nueve diagramas estándares del UML: diagrama de casos de uso, de clases, de secuencia, de colaboración, de actividad, de estados, de implementación (componentes), de despliegue y varios perfiles del UML. Es muy flexible y completa. Desde los inicios del proyecto de Video Vigilancia SURIA Enterprise Architect fue la herramienta autorizada por el arquitecto para utilizar durante su realización. Posee todas las funcionalidades del Visual Paradigm, gran velocidad, buen rendimiento y estabilidad.

1.13. Conclusiones parciales

Este capítulo sirvió de base para la elaboración de un sistema donde la herramienta y metodología seleccionadas constituyen la fuerza fundamental para su desarrollo exitoso. El estudio de todos los software existentes con funcionalidades similares a las requeridas por el producto a desarrollar, así como el estudio de los distintos temas relacionados con los sistemas de grabación de video, ayudan a obtener un conjunto de conocimientos que convergen en lo que será el correcto análisis del sistema.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción

Con motivo de la poca estructuración de los procesos del negocio y para poder comprender el contexto en el cual se desarrolla el sistema se determinó desarrollar un Modelo de Dominio, donde se expone un marco conceptual y las relaciones entre estas definiciones. Por otra parte, se enumeran los requerimientos funcionales y no funcionales, agrupándose los primeros en Casos de Uso, con el fin de estructurar el Diagrama de Casos de Uso del Sistema. Se definen los patrones de caso de uso que serán utilizados, además se realiza el diagrama de clases del análisis de cada uno de estos casos de uso así como el diagrama de colaboración de cada una de estas clases.

2.2. Modelo de dominio

El Modelo de Dominio o Modelo Conceptual, permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Un Modelo del Dominio es una representación de las clases conceptuales del mundo real, no de componentes software. El modelo desarrollado no se trata de un conjunto de diagramas que describen clases de software u objetos de software con responsabilidades, sino que puede considerarse como un diccionario visual de las abstracciones relevantes, vocabulario e información del dominio. Aprovechando las bondades de los diagramas UML para representar conceptos, el modelo de dominio se presenta en forma de diagrama de clases donde figuran los principales conceptos y roles del sistema en cuestión. (16)

2.2.1. *Conceptos fundamentales*

Para un mejor entendimiento del diagrama de modelo de dominio conformado, en este punto se proporciona un marco conceptual con las definiciones identificadas, las cuales son:

Cámara: Dispositivo de captura de imagen y video, convirtiendo estos en señales eléctricas que pueden ser reproducidos por un aparato determinado.

Video: Archivo que almacena los flujos de video de una cámara, comprimidos en un formato de uso común.

Calendario: Conjunto de reglas de grabación definidas para una cámara.

Grabación: Es un proceso de grabación para una cámara en un momento determinado.

Regla de grabación: Define en qué momento se graba desde una cámara y durante qué espacio de tiempo. Es repetitiva pudiendo tener una frecuencia mensual, semanal o diaria.

Tarea de grabación: Define en qué momento se graba desde una cámara especificando la fecha y hora de inicio y la fecha y hora de fin.

2.2.2. Diagrama del modelo de dominio

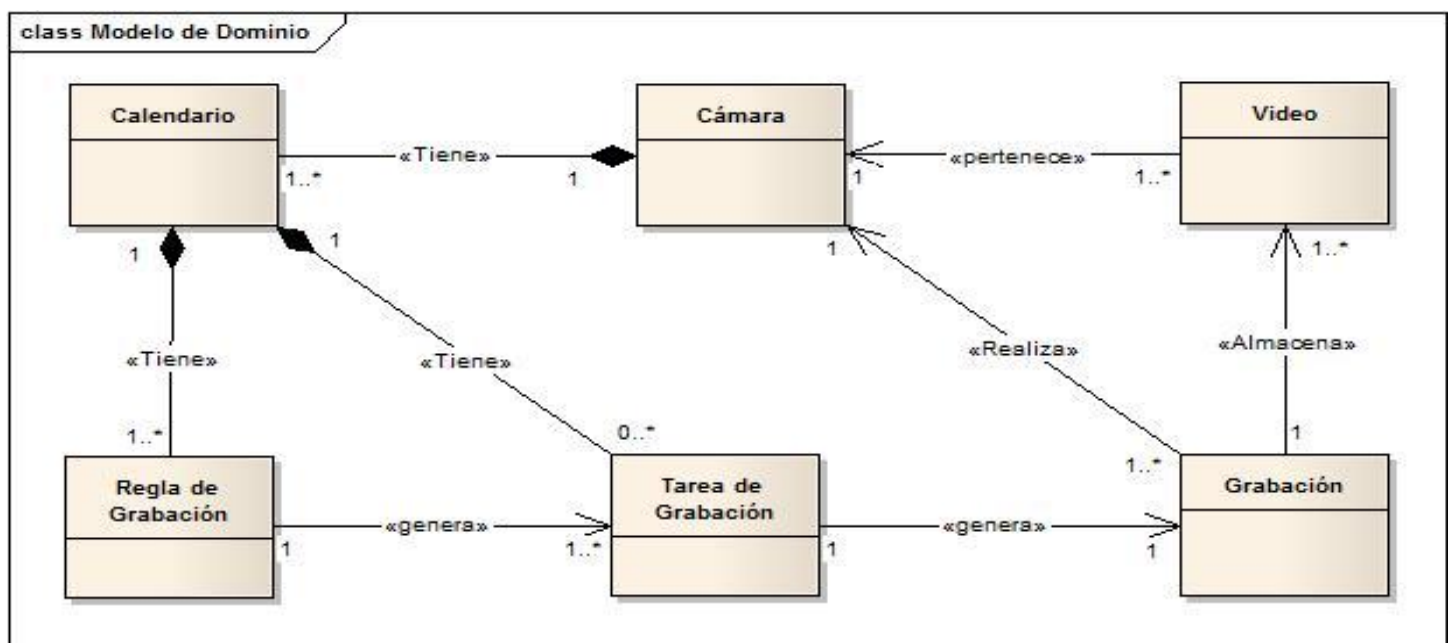


Figura 1: Modelo de dominio

2.3. Requerimientos funcionales del sistema

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

2.3.1. Requerimientos funcionales del servidor de grabación

RF1 Grabar flujo de video desde cámara IP (Grabar).

RF 1.1 Grabar de forma manual.

RF 1.2 Grabar mediante un calendario.

RF 1.3 Grabar ante la ocurrencia de un evento.

RF 2 Cargar plugins de grabación.

RF 3 Almacenar los videos (siguiendo una estructura lógica que describa la cámara a la que pertenece, y el rango de tiempo que comprende).

RF 4 Funcionar de modo autónomo.

RF 4.1 Ante un fallo en un proceso de grabación, detectarlo y reiniciarlo cuando sea posible.

RF 4.2 Detectar la caída de la conexión con el Gestor y restablecerla cuando vuelva a estar disponible.

RF 5 Brindar información al Gestor.

2.3.2. Requerimientos funcionales de cada plugin de grabación

RF 6 Grabar desde uno o varios tipos de cámaras.

RF 7 Ejecutar tarea de grabación en medio controlado.

RF 8 Detectar fallo en el proceso de grabación.

RF 8.1 Detectar caída de la conexión con la cámara.

RF 8.2 Detectar bloqueo del proceso externo de grabación.

RF 9 Brindar estado del proceso de grabación.

2.3.3. Requerimientos funcionales del cliente

RF 10 Mostrar cámaras existentes.

RF 11 Mostrar calendario de grabación.

RF 11.1 Mostrar calendario en forma de listas (de reglas y tareas de grabación).

RF 11.2 Mostrar calendario de grabación en forma de gráfica.

RF 12 Gestionar calendario de grabación.

RF 12.1 Adicionar regla de grabación mediante wizard⁷.

RF 12.2 Eliminar regla de grabación.

RF 12.3 Modificar regla de grabación mediante wizard.

RF 12.4 Modificar calendario de grabación en forma gráfica.

RF 13 Iniciar o detener grabación manualmente.

RF 14 Configurar cliente.

2.3.4. *Requerimientos funcionales del editor de configuración*

RF 15 Editar configuración del servidor de grabación.

RF 16 Iniciar o detener la conexión con el servidor de grabación.

2.4. Requerimientos no funcionales del sistema

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*

Nota: Se usa el prefijo **RNU** en su nomenclatura.

RNU 1 La interfaz debe ser configurable para lograr la comodidad del usuario.

RNU 2 El cliente⁸ 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.

⁷ Wizard: Es una interfaz de usuario elemental, constituida por una secuencia de ventanas que guían al usuario a través de una serie de pasos bien definida.

⁸ Cliente: Refiriéndose a la aplicación cliente del Grabador.

RNU 4 El Sistema debe mostrar mensajes al usuario, que le ayuden a llevar a cabo la tarea que realiza.

RNU 5 Se debe hacer uso de botones, con imágenes que indiquen de modo intuitivo la función que realizan.

RNU 6 Los controles visuales deben mostrar mensajes que indiquen su función.

RNU 7 La configuración predeterminada debe brindar buena comodidad.

RNU 8 Los parámetros de funcionamiento del módulo deben ser configurables.

2.4.2. Requerimientos de fiabilidad

Nota: Se usa el prefijo **RNF** en su nomenclatura.

RNF 9 El sistema debe estar disponible de forma permanente.

RNF 10 Debe funcionar sin necesidad de la intervención del usuario.

2.4.3. Requerimientos de eficiencia

Nota: Se usa el prefijo **RNE** en su nomenclatura.

RNE 11 Debe usar programación concurrente para lograr un óptimo aprovechamiento de los recursos de hardware.

RNE 12 El sistema debe poder distribuir las tareas de grabación entre varias instancias del Servidor de Grabación, corriendo en diferentes máquinas para balancear la carga de trabajo.

2.4.4. Requerimientos de diseño e implementación

Nota: Se usa el prefijo **RNDI** en su nomenclatura.

RNDI 13 Debe ser implementado en C++ ajustándose a las funcionalidades del Framework de desarrollo QT.

RNDI 14 Resolución de 1024 x 768.

2.4.5. *Requerimientos de soporte*

Nota: Se usa el prefijo **RNSO** en su nomenclatura.

RNSO 15 En caso de que el cliente adquiriera un nuevo modelo de cámara, se le debe poder suministrar un plugin de grabación que lo soporte, en caso de que los actuales no lo hagan.

2.4.6. *Requerimientos de interfaz de usuario*

Nota: Se usa el prefijo **RNIU** en su nomenclatura.

RNIU 16 Se requiere que el cliente tenga una interfaz gráfica que permita la interacción con el usuario.

RNIU 17 Se requiere una interfaz ligera, que permita de manera sencilla configurar el servidor de grabación.

RNIU 18 Se requiere una interfaz gráfica que permita administrar el espacio usado para el almacenamiento.

2.4.7. *Requerimientos de interconexión*

Nota: Se usa el prefijo **RNI** en su nomenclatura.

RNI 19 Debe poder comunicarse con cualquier aplicación que implemente las interfaces definidas para el Gestor, asegurando que este pueda ser actualizado o sustituido sin afectar el funcionamiento del Grabador.

RNI 20 Debe brindar una interfaz fija para la comunicación con sistemas externos que usen las funciones que el Grabador brinda.

2.4.8. *Requerimientos de funcionamiento*

Nota: Se usa el prefijo **RNFO** en su nomenclatura.

RNFO 21 Debe ser multiplataforma, es decir que funcione para los sistemas operativos Windows (XP, SP2 o superior) y Linux.

RNFO 22 Debe tener instalados Códec de video.

RNFO 23 Giga bit Ethernet NIC (recomendado).

RNFO 24 Tarjeta de video (recomendado).

2.4.9. *Requerimientos de seguridad*

Nota: Se usa el prefijo **RNS** en su nomenclatura.

RNS 25 Se garantizará la seguridad del Sistema mediante la autenticación de usuarios.

RNS 26 Se registrarán logs de las acciones llevadas a cabo en el sistema.

2.5. Definición de los actores y casos de uso del sistema

2.5.1. *Actores del sistema*

Actor	Descripción
Gestor.	Es el actor del sistema que tiene permiso para gestionar al sistema externo.
Usuario.	Es el actor del sistema que representa a una persona que interactúe con el sistema antes de autenticarse
Usuario con privilegios.	Es el actor del sistema que representa a un usuario luego de autenticarse en el sistema cuyos privilegios son asignados de acuerdo al esquema de seguridad establecido. De estos privilegios dependerán las acciones que podrá llevar a cabo dentro del sistema.
Temporizador.	Es un actor ficticio del sistema que representa un intervalo de tiempo tras el cual se realiza una tarea de manera repetitiva.
Servidor de grabación.	Es el actor del sistema que representa a una aplicación que usa al plugin de grabación.

Editor de configuración.	Es el actor del sistema que representa a una aplicación a través de la cual el usuario con privilegios puede modificar los datos de configuración del servidor de grabación.
---------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2.5.2. *Listado de los casos de uso del servidor de grabación*

Caso de Uso 1:	Cargar plugins de grabación.
Actores:	Editor de configuración.
Descripción:	Permite cargar los plugins de grabación disponibles.
Referencias:	RF 2.

Caso de Uso 2:	Configurar.
Actores:	Editor de configuración.
Descripción:	Permite cargar la configuración del servidor de grabación así como los Plugins de Grabación desde bibliotecas de clases externas.
Referencias:	RNU 8.

Caso de Uso 3:	Grabar.
Actores:	Gestor.
Descripción:	Permite crear un proceso de grabación para una cámara determinada usando un plugin de grabación adecuado y comenzar o detener dicho proceso.

Referencias:	RF 1, RF 3.
---------------------	-------------

Caso de Uso 4:	Brindar información.
Actores:	Gestor.
Descripción:	Permite brindarle información al Gestor acerca de las tareas de grabación que se están ejecutando, la memoria virtual disponible y la carga de CPU.
Referencias:	RF 5.

Caso de Uso 5:	Monitorizar.
Actores:	Temporizador.
Descripción:	Permite chequear de forma continua el estado de la conexión con el Gestor y de los procesos de grabación.
Referencias:	RF 4.

Caso de Uso 6:	Registrar log.
Actores:	Gestor.
Descripción:	Permite que el servidor de grabación archive <i>logs</i> de cada una de las operaciones realizadas.
Referencias:	RNS 27.

Caso de Uso 7:	Conectarse al sistema externo.
Actores:	Editor de configuración.

Descripción:	Permite establece la conexión con el Gestor.
Referencias:	RNU 8.

Listado de los casos de uso del cliente

Caso de Uso 8:	Autenticar usuario.
Actores:	Usuario.
Descripción:	Garantizar que la persona que esté operando el sistema tenga los derechos requeridos.
Referencias:	RNS 26.

Caso de Uso 9:	Configurar cliente.
Actores:	Usuario con privilegios.
Descripción:	Permite modificar los parámetros de configuración del cliente.
Referencias:	RF 14.

Caso de Uso 10:	Gestionar calendarios de grabación.
Actores:	Usuario con privilegios.
Descripción:	Permite definir reglas para la grabación, aplicadas a cada cámara. Las cuales consisten en indicar con qué frecuencia se graba (mensual, semanal o diario), la hora de inicio y la duración o la hora de fin a elección del usuario, agregar reglas de grabación, modificar reglas de grabación y eliminar reglas de grabación.

Referencias:	RF 12.
---------------------	--------

Caso de Uso 11:	Grabar manualmente.
Actores:	Usuario con privilegios.
Descripción:	Permite que un usuario inicie un proceso de grabación para una cámara determinada.
Referencias:	RF 13.

Caso de Uso 12:	Mostrar cámaras instaladas.
Actores:	Usuario con privilegios.
Descripción:	Permite visualizar el conjunto de cámaras manejadas por el sistema, agrupadas en una jerarquía atendiendo a su ubicación física.
Referencias:	RF 10.

Caso de Uso 13:	Registrar log.
Actores:	Usuario con privilegios.
Descripción:	Permite que el cliente archive <i>logs</i> de cada una de las operaciones realizadas.
Referencias:	RNS 27.

Caso de Uso 14:	Visualizar calendario de grabación.
Actores:	Usuario con privilegios.
Descripción:	Permite mostrar las reglas y tareas de grabación, ya sea como listas de reglas y tareas de grabación o como un gráfico de tiempo indicando los espacios de tiempo en los que se graba y en los que no.
Referencias:	RF 11.

2.5.3. *Listado de los casos de uso del editor de configuración*

Caso de Uso 15:	Autenticar usuario.
Actores:	Usuario.
Descripción:	Garantizar que la persona que esté operando el sistema tenga los derechos requeridos.
Referencias:	RNS 26.

Caso de Uso 16:	Configurar servidor de grabación.
Actores:	Usuario con privilegios.
Descripción:	Permite modificar los parámetros de configuración del servidor de grabación.
Referencias:	RF 15.

Caso de Uso 17:	Registrar log.
Actores:	Usuario con privilegios.
Descripción:	Permite que el editor de configuración archive <i>logs</i> de cada una de las operaciones realizadas.
Referencias:	RNS 27.

Caso de Uso 18:	Manejar servidor de grabación.
Actores:	Usuario con privilegios.
Descripción:	Permite iniciar o detener el servidor de grabación.
Referencias:	RF 16.

2.5.4. *Listado de los casos de uso del plugin de grabación*

Caso de Uso 19:	Grabar.
Actores:	Servidor de grabación.
Descripción:	Permite grabar los flujos de video de una cámara en uno o varios archivos de video.
Referencias:	RF 6, RF 7.

Caso de Uso 20:	Monitorizar proceso de grabación.
Actores:	Temporizador.

Descripción:	Permite chequear de forma continua el estado de la grabación, y controlar la duración máxima de cada archivo de video.
Referencias:	RF 8.

Caso de Uso 21:	Notificar eventos.
Actores:	Servidor de grabación.
Descripción:	Permite notificar a la aplicación servidor de grabación los eventos ocurridos en el proceso de grabación.
Referencias:	RF 9.

Caso de Uso 22:	Inicializar plugin
Actores:	Servidor de grabación.
Descripción:	Permite comenzar el proceso de monitoreo.
Referencias:	RF 6 y RF 7.

2.6. Patrones para la identificación de los casos de uso del sistema

A medida que ha ido evolucionando la utilización de casos de uso también se han ido desarrollando una serie de patrones que facilitan tener más precisión a la hora de reflejar los requisitos reales, haciendo más fácil el trabajo con los sistemas, y mucho más simple su mantenimiento. Dado un contexto y un problema a resolver, estas técnicas han mostrado ser la solución adoptada en la comunidad del desarrollo de software. Se presentan a modo de herramientas que permiten resolver los problemas que se les planteen a los desarrolladores de una forma ágil y sistemática. Estos patrones se enfocan hacia el diseño y las técnicas utilizadas en modelos de alta calidad, y no en cómo modelar usos específicos. (17)

Dentro de los principales patrones se encuentran: Concordancia y CRUD (Creating, Reading, Updating, Deleting) los cuales serán utilizados en el desarrollo de la solución propuesta.

2.6.1. Concordancia

Extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado. Además se puede representar de dos formas distintas: Reusabilidad y Adición. (17) En el tema a tratar se puede observar en las siguientes figuras (Figura 2, Figura 3).



Figura 2: Concordancia (reusabilidad)

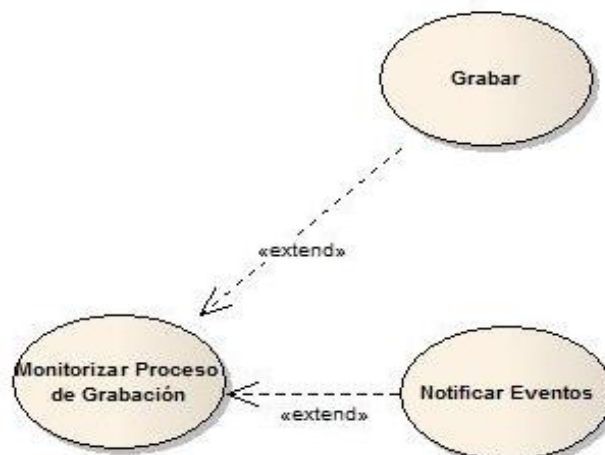


Figura 3: Concordancia (adición)

2.6.2. CRUD (Creating, Reading, Updating, Deleting)

Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual. Se puede representar de dos formas distintas: Completo el cual modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación y Parcial el cual modela una de las vías de los casos de uso como un caso de uso separado. (17) En el tema a tratar se puede observar el Completo en la siguiente figura (Figura 4).

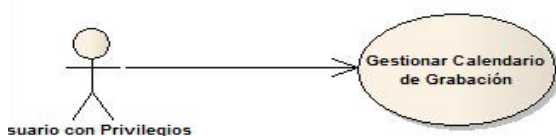


Figura 4: CRUD (completo)

2.7. Diagramas de casos de uso del sistema

2.7.1. Diagrama de casos de uso del servidor de grabación

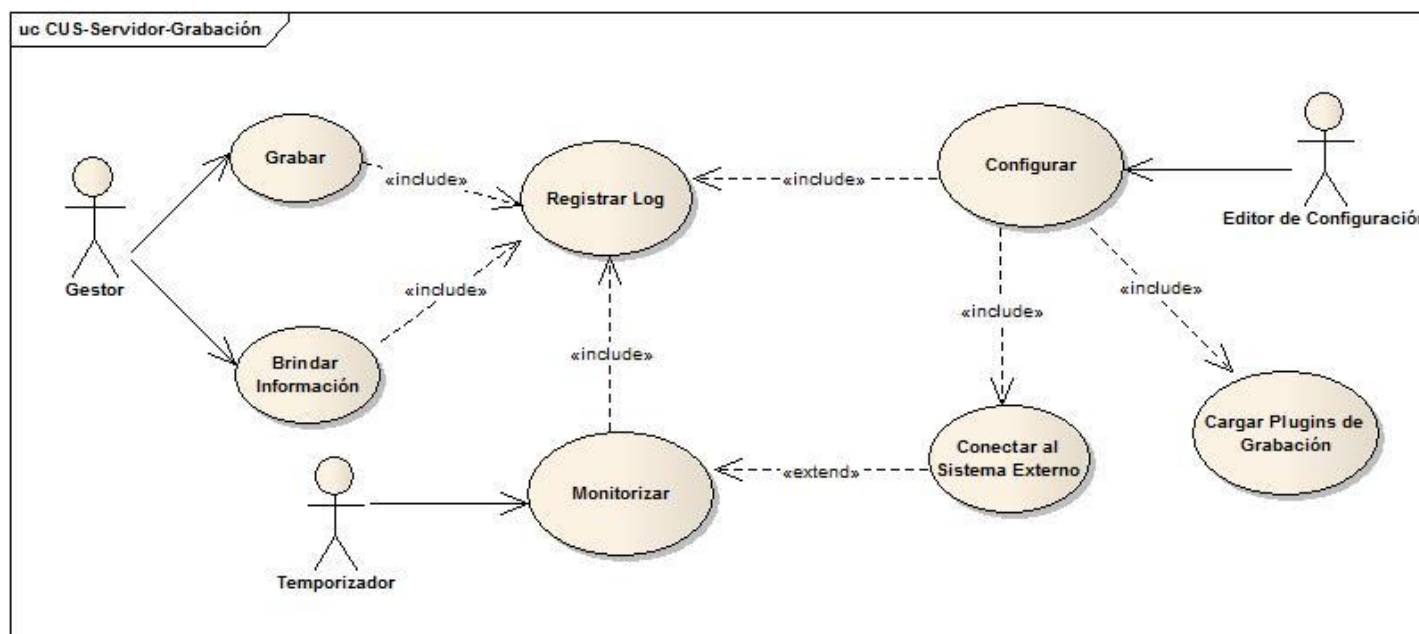


Figura 5: Diagrama de casos de uso del servidor de grabación

2.7.2. Diagrama de casos de uso del cliente

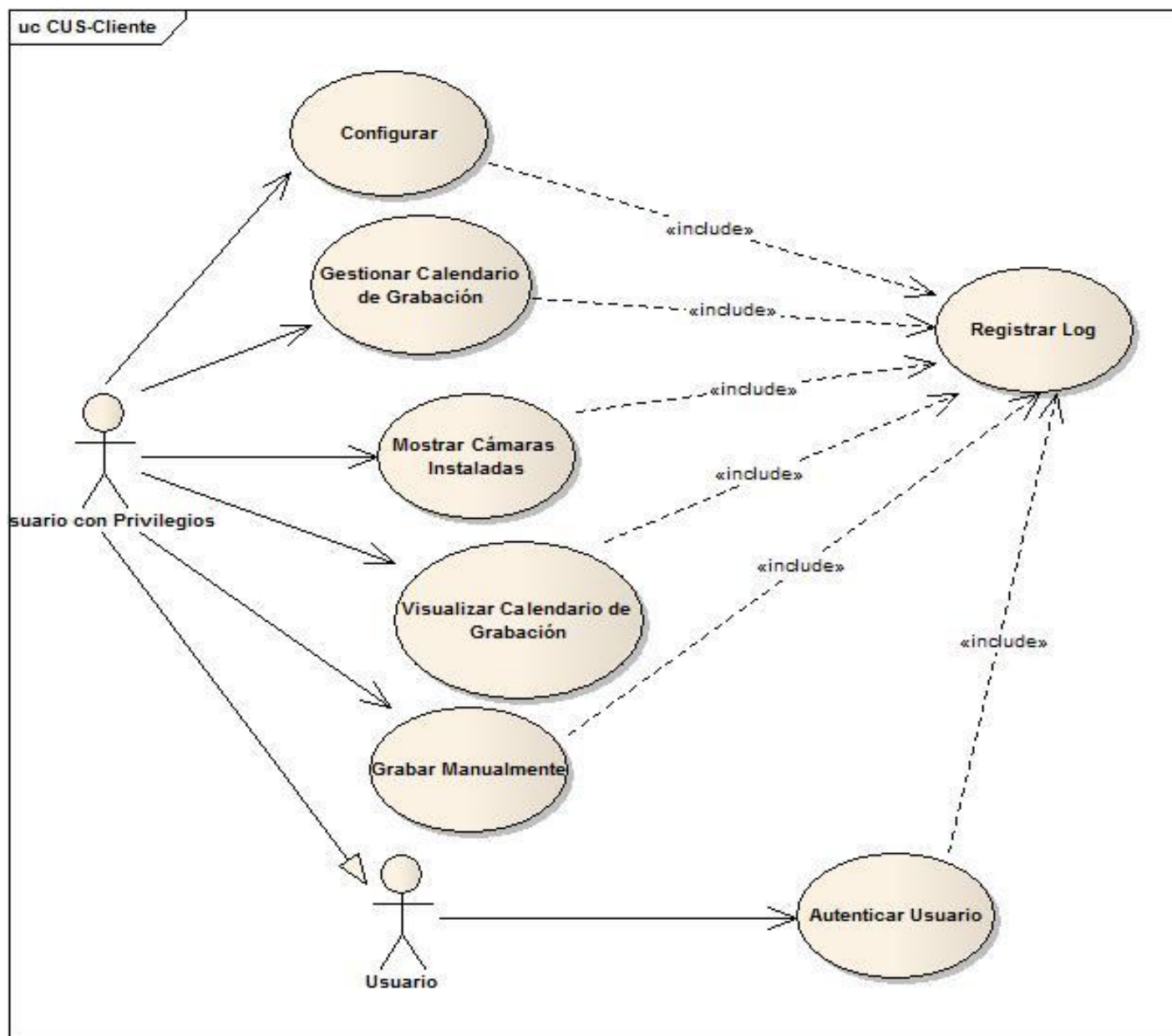


Figura 6: Diagrama de casos de uso del cliente

2.7.3. Diagrama de casos de uso del editor de configuración

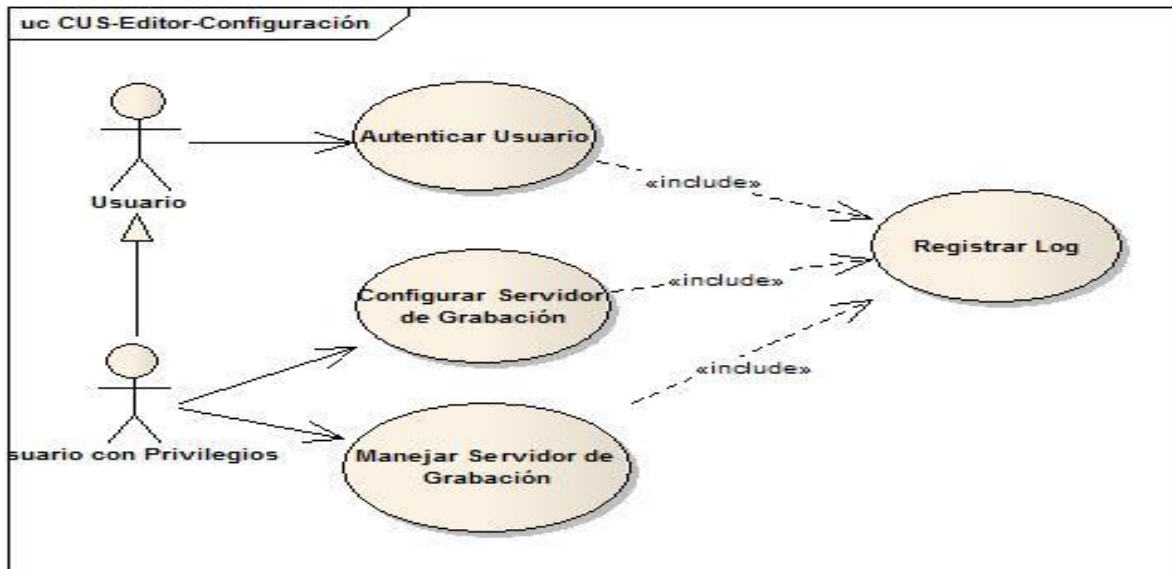


Figura 7: Diagrama de casos de uso del editor de configuración

2.7.4. Diagrama de casos de uso del plugin de grabación

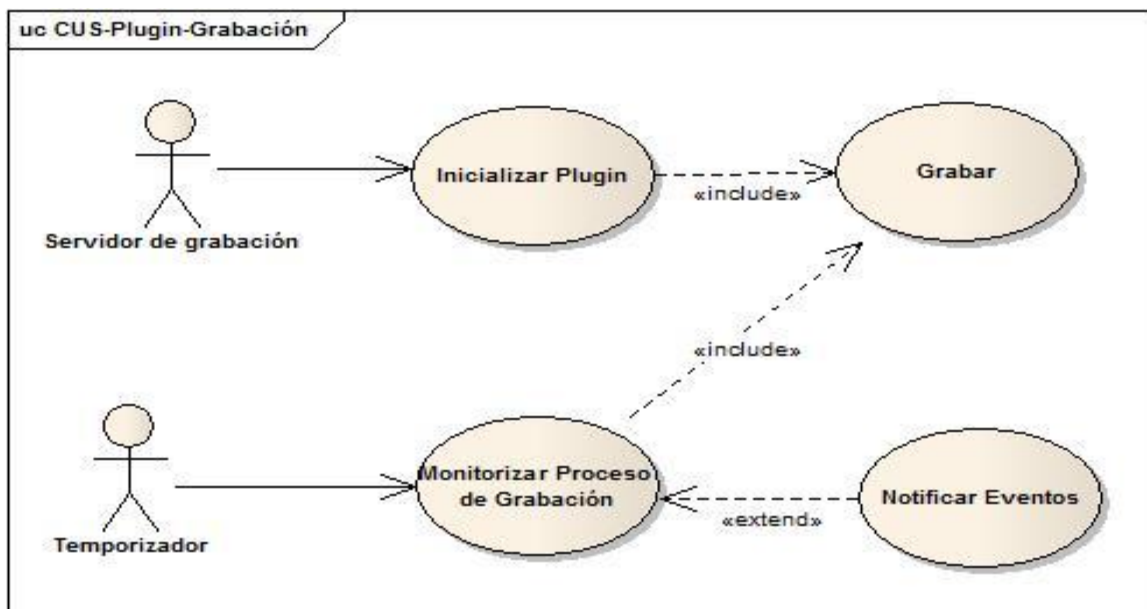


Figura 8: Diagrama de casos de uso del plugin de grabación

2.8. Modelo de análisis

El análisis consiste en obtener una visión del sistema que se preocupa de lo que hace el mismo. Es la entrada fundamental para el comienzo del diseño.

2.8.1. Diagramas de clases del análisis del servidor de grabación

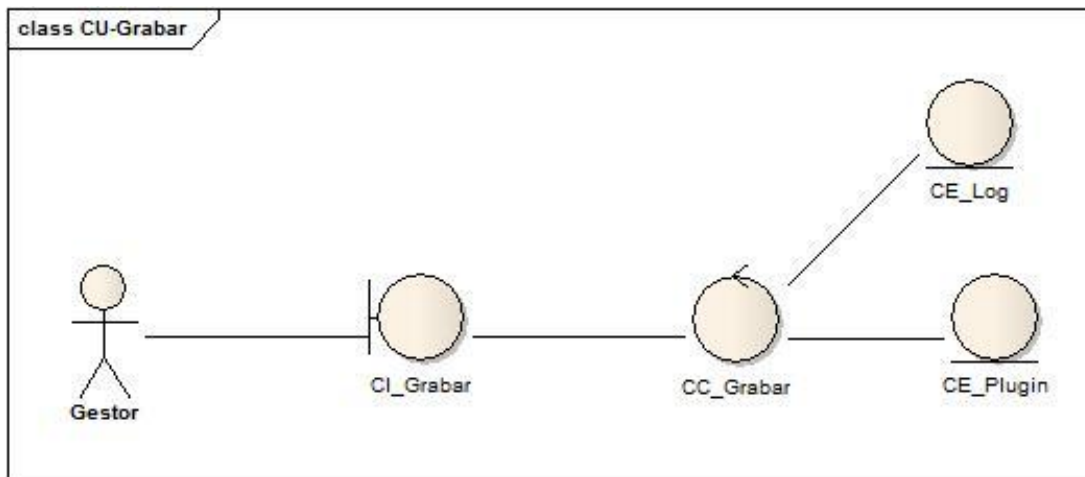


Figura 9: CU-Grabar

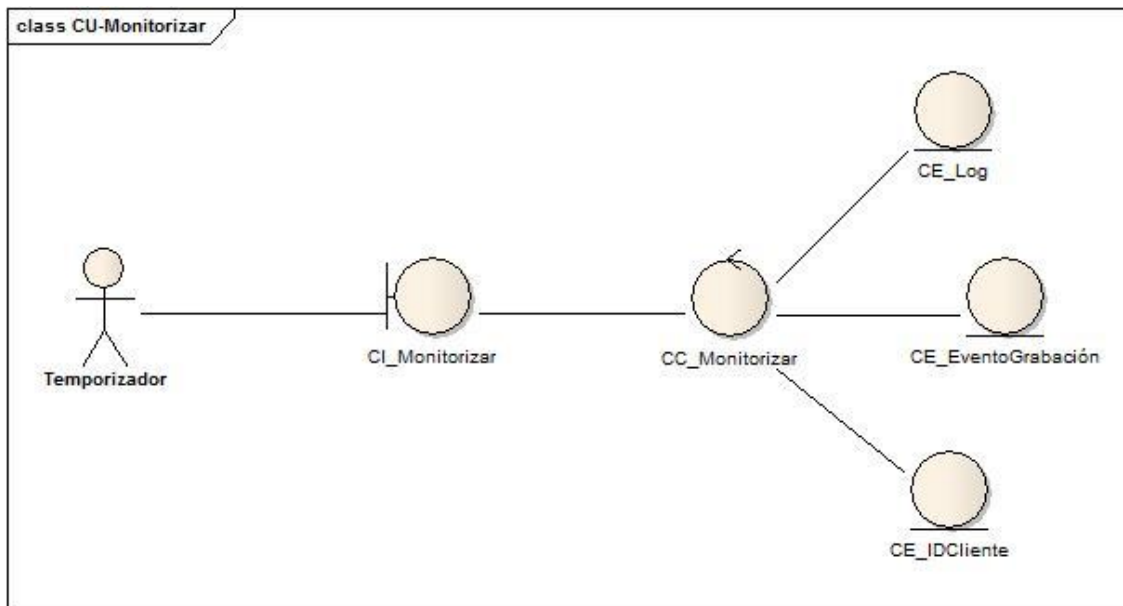


Figura 10: CU-Monitorizar

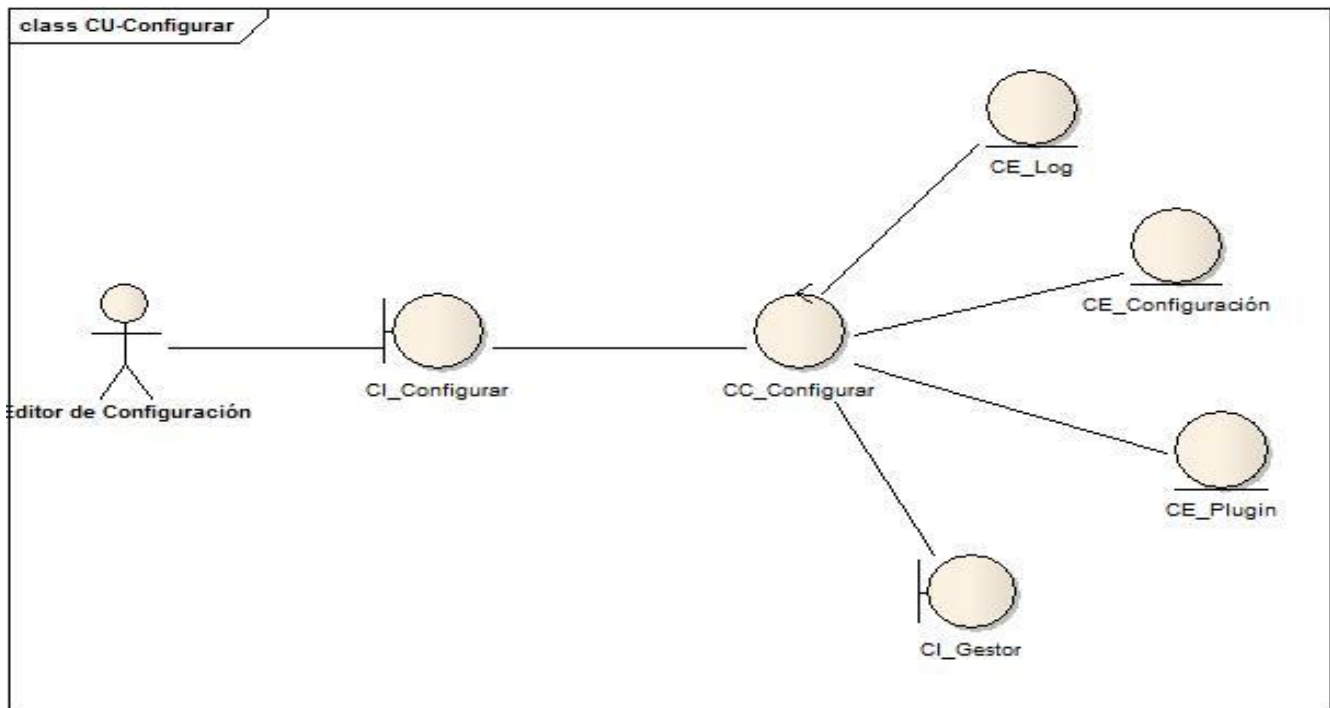


Figura 11: CU-Configurar

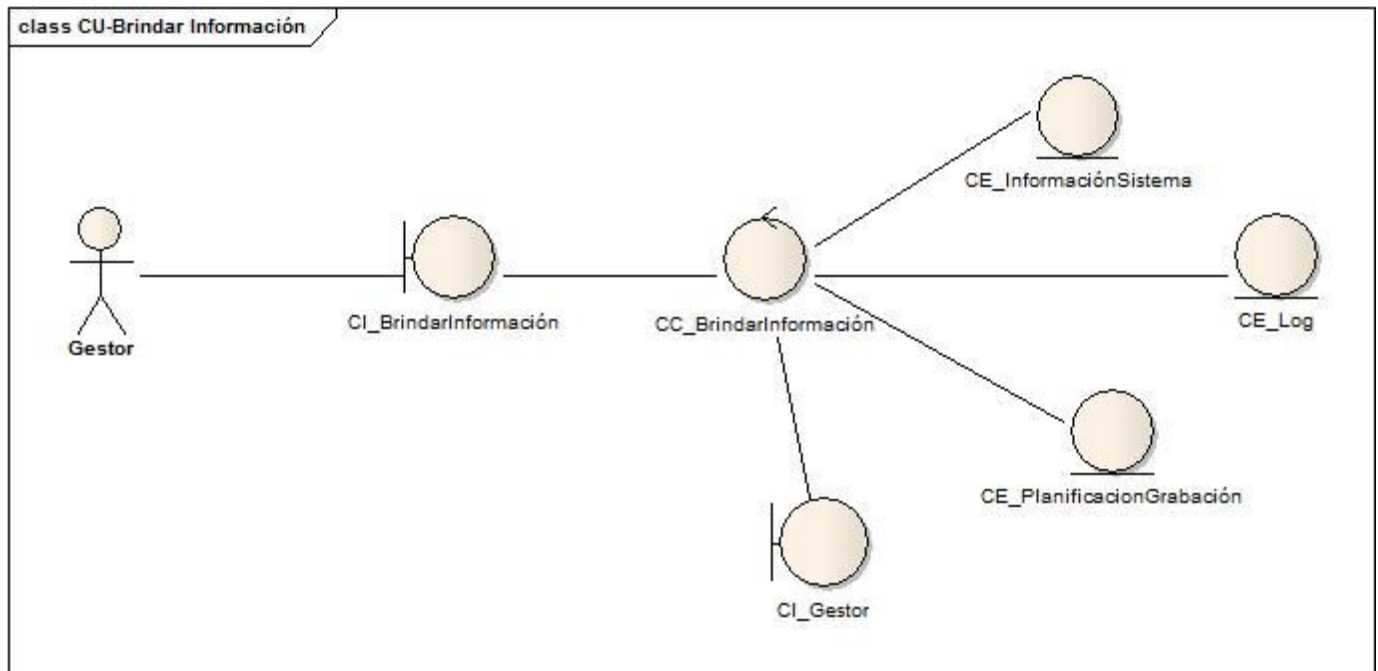


Figura 12: CU-Brindar información

2.8.2. Diagramas de clases del análisis del cliente

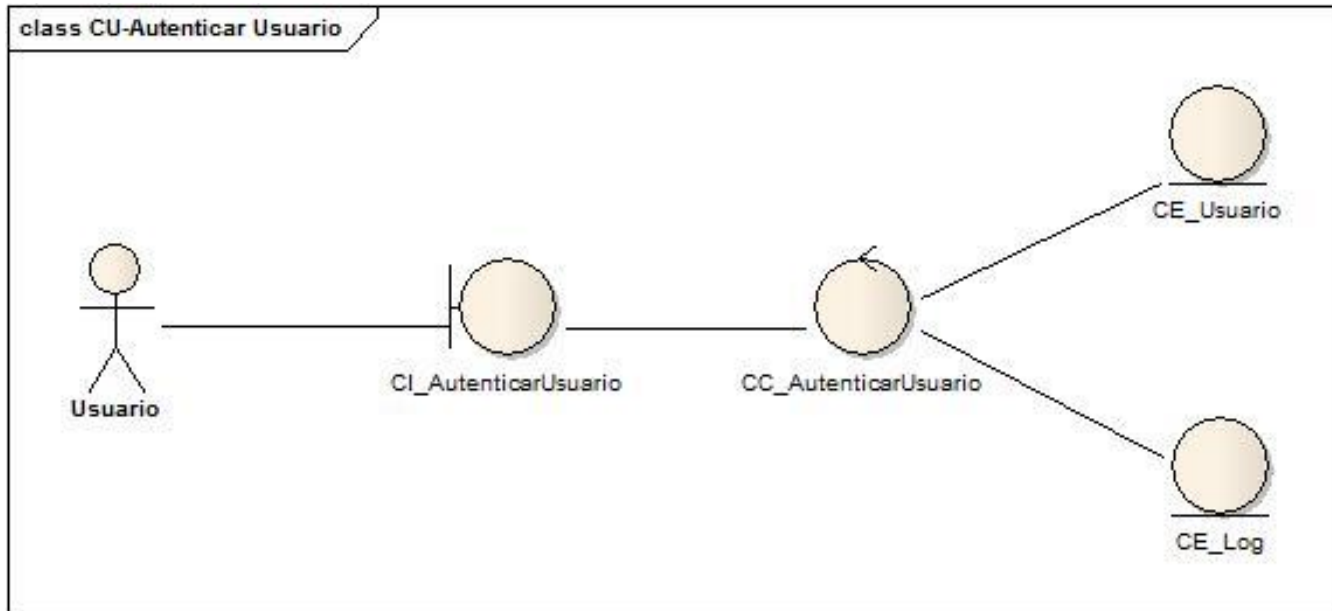


Figura 13: CU-Autenticar usuario

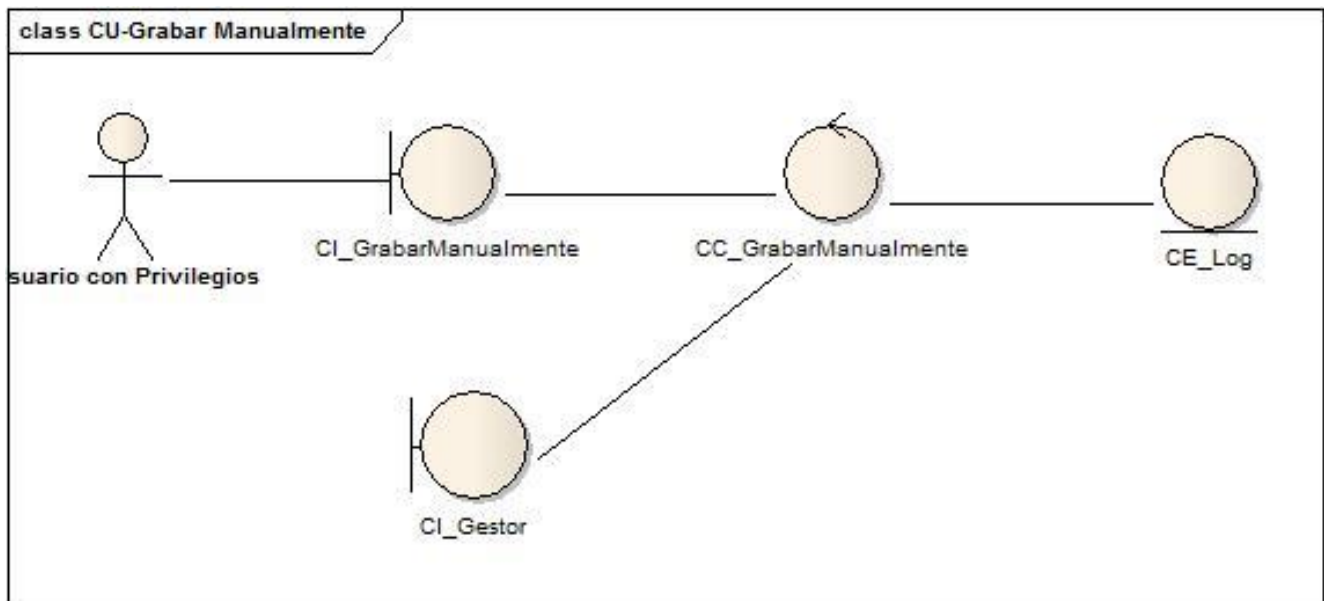


Figura 14: CU-Grabar manualmente

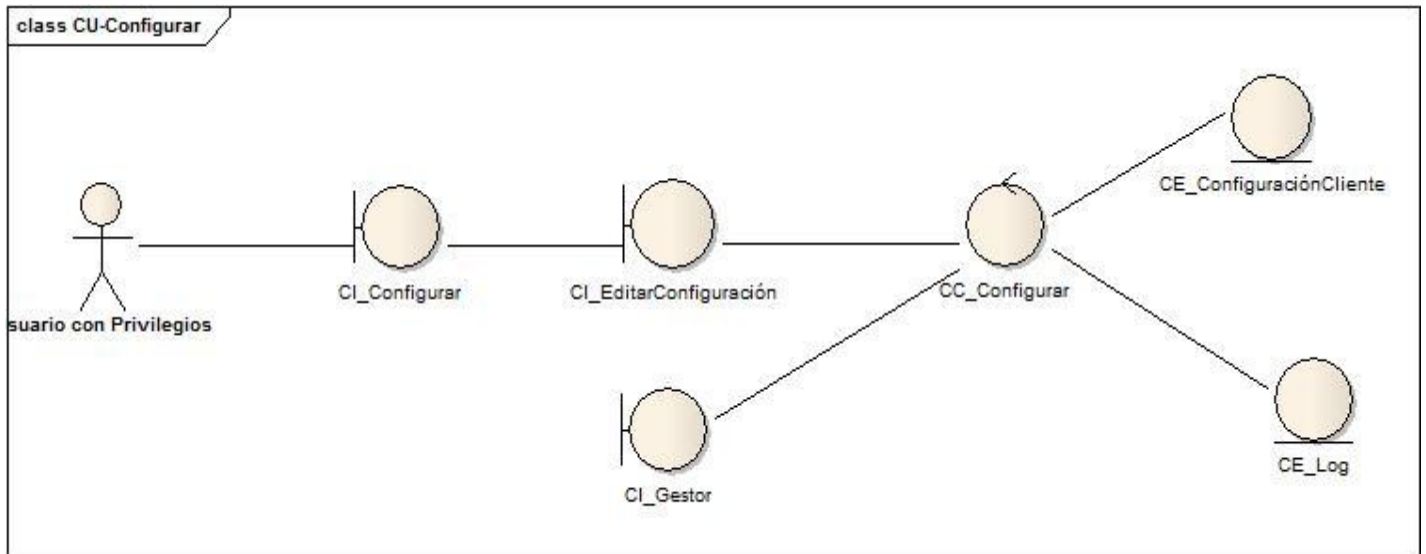


Figura 15: CU-Configurar

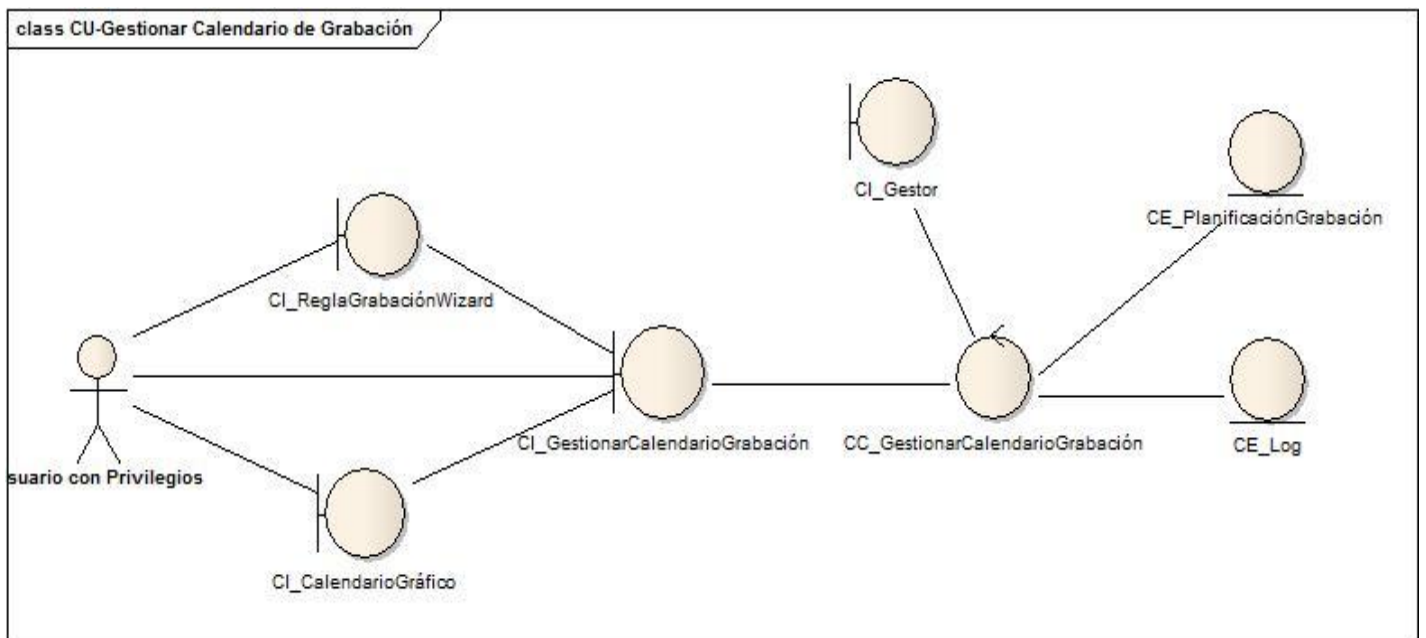


Figura 16: CU-Gestionar calendario de grabación

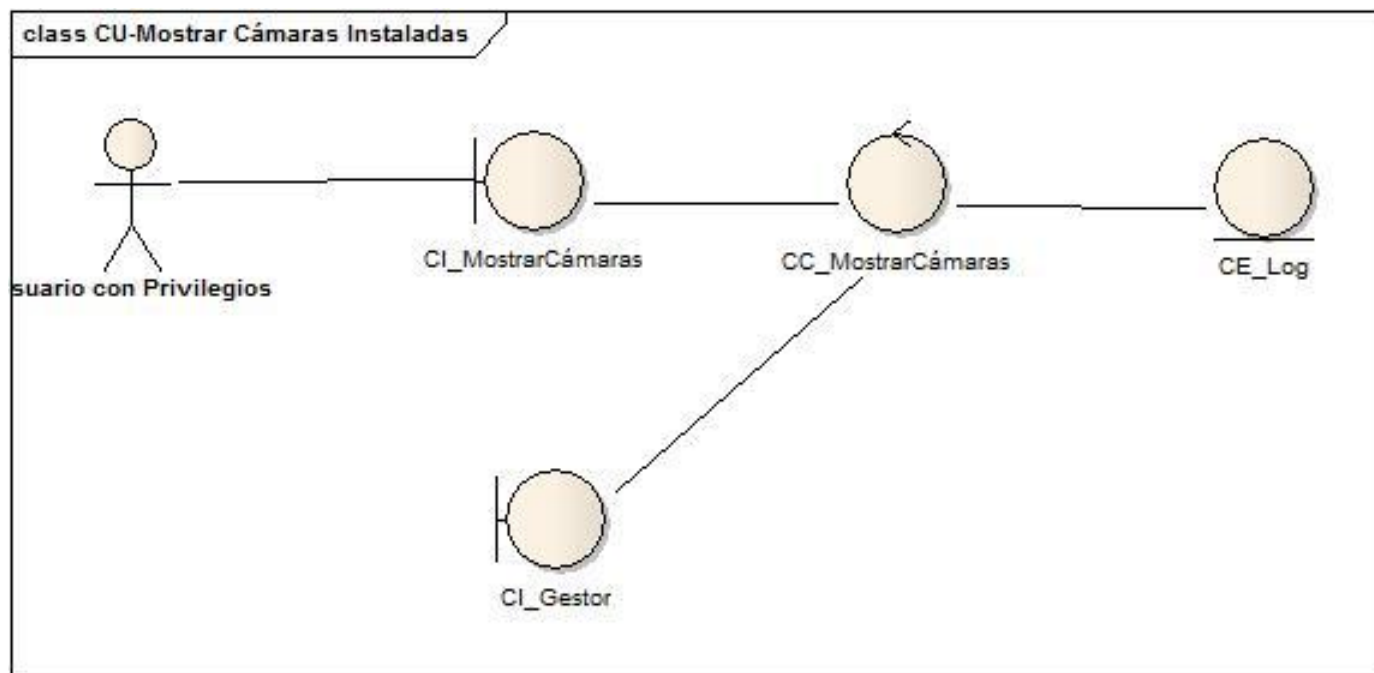


Figura 17: CU-Mostrar cámaras instaladas

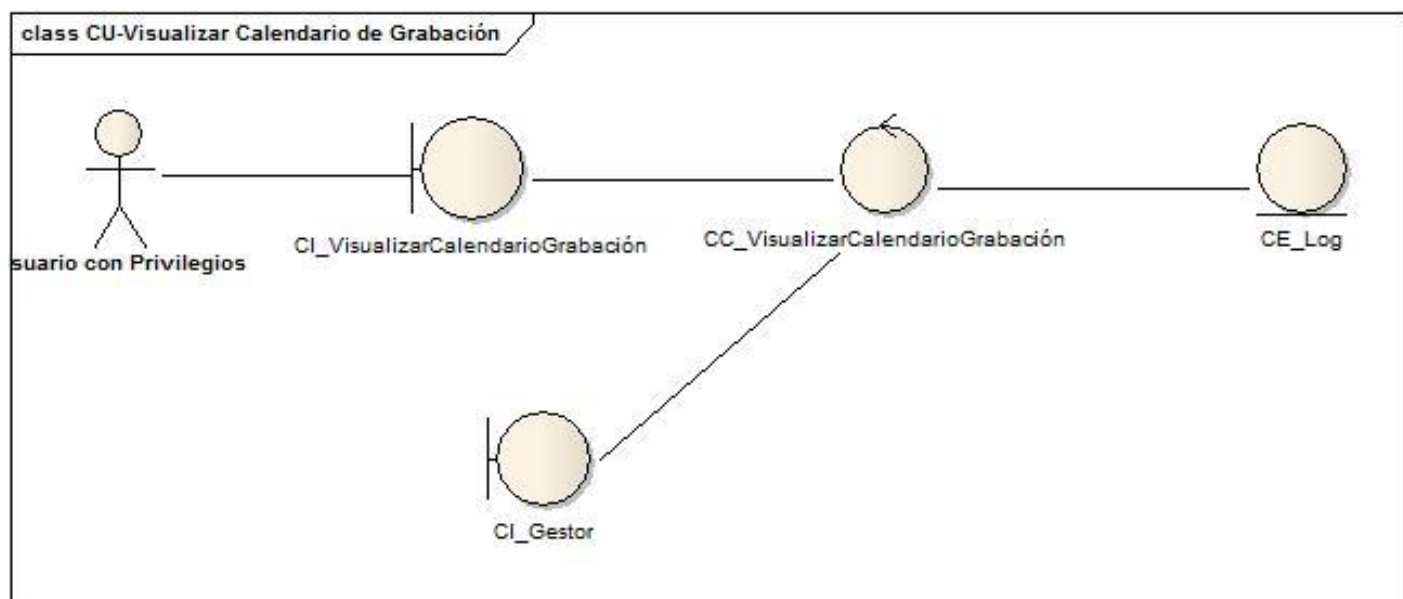


Figura 18: CU-Visualizar calendario de grabación

2.8.3. Diagramas de colaboración de servidor de grabación

CU-Grabar

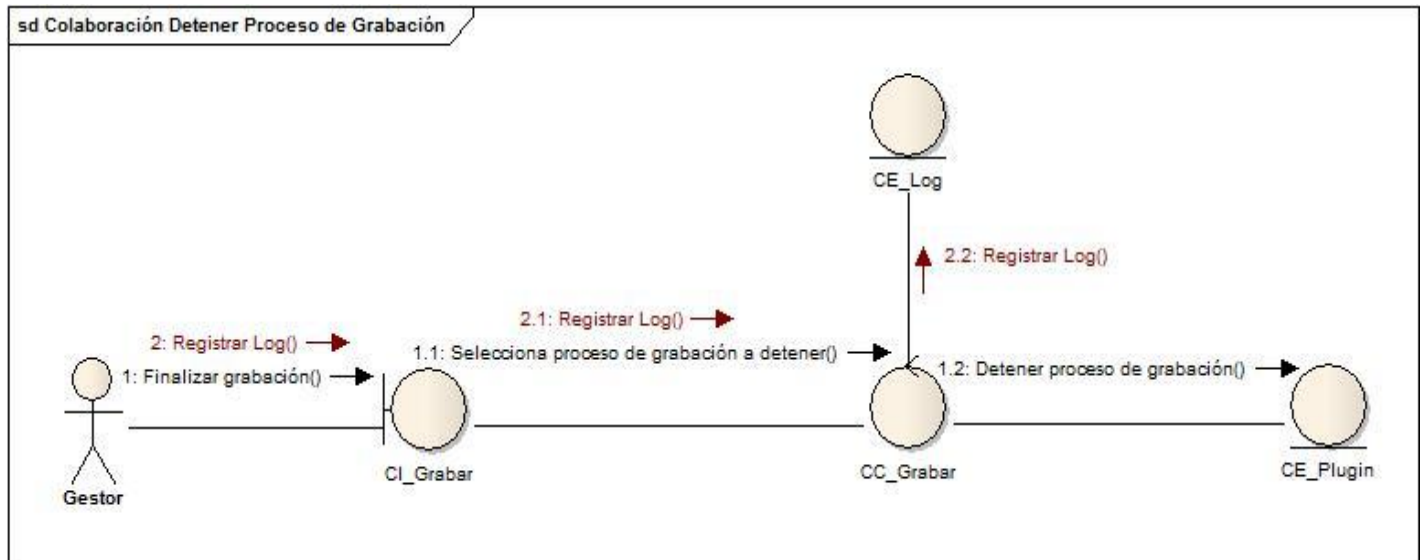


Figura 19: Escenario detener proceso de grabación

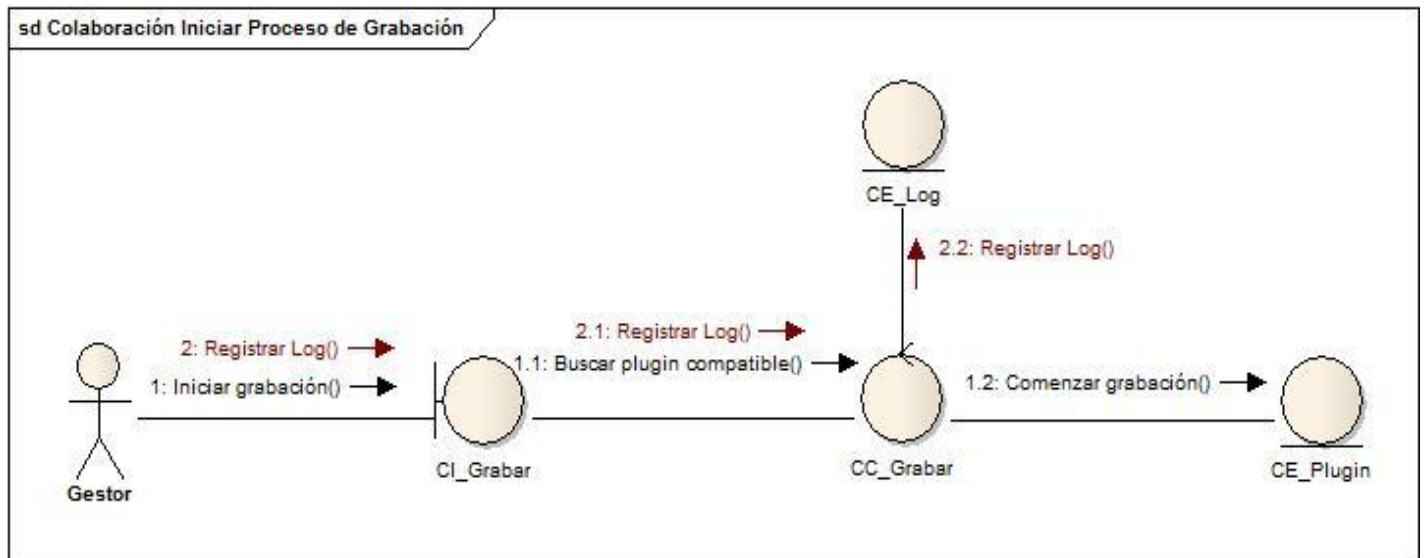


Figura 20: Escenario iniciar proceso de grabación

2.8.4. Diagramas de colaboración del cliente

CU-Autenticar usuario

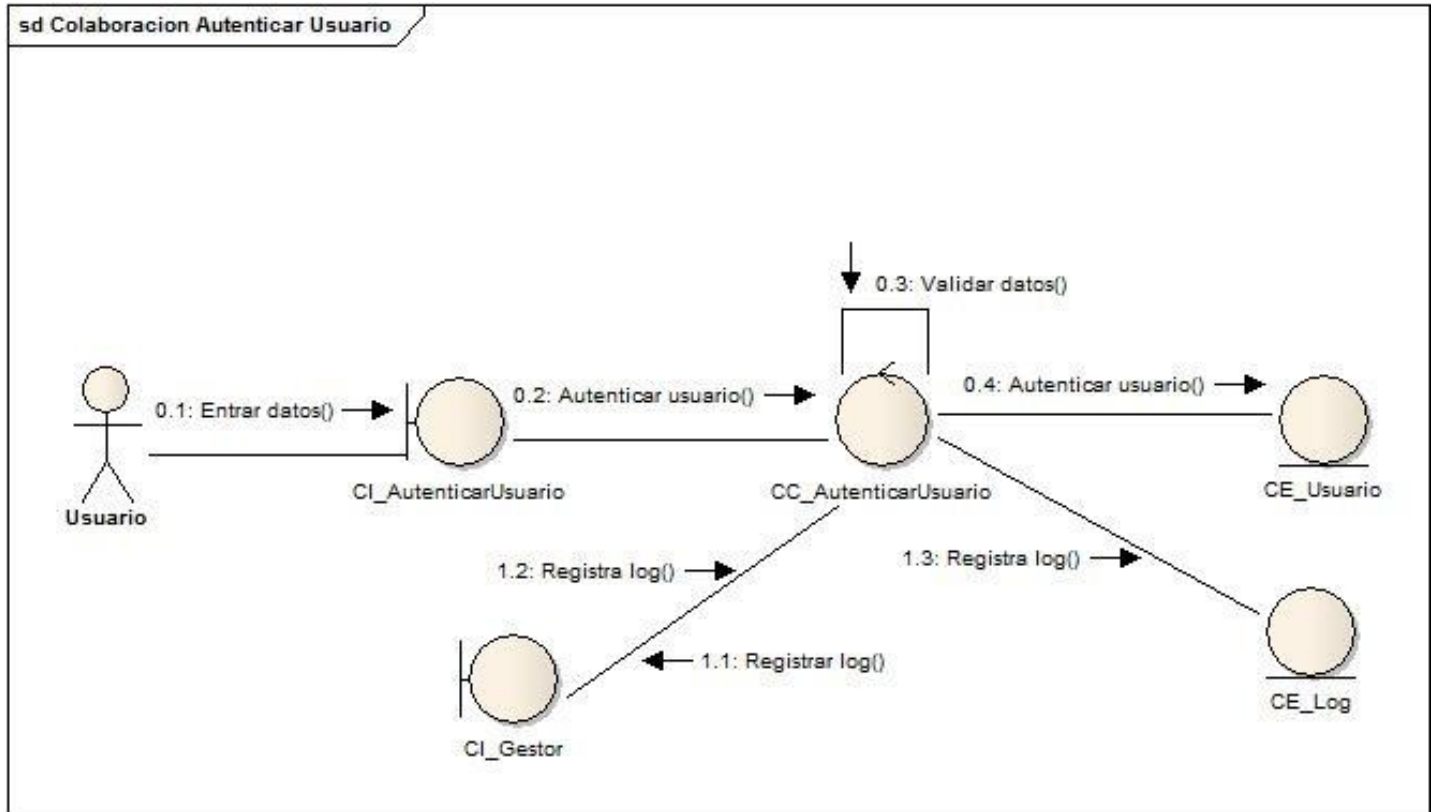


Figura 21: Autenticar usuario

2.9. Conclusiones parciales

Con la realización de este capítulo se obtuvo un mayor entendimiento del proceso de negocio, definiendo así los requisitos funcionales y no funcionales que permitieron identificar las funcionalidades con las que contará el sistema, las cuales darán respuesta a las necesidades del problema. Los casos de uso fueron relacionados mediante los diagramas de casos de uso del sistema y se les realizó una descripción detallada logrando así un mayor acercamiento a lo que el sistema debería cumplir. Por último se expuso el modelo de análisis a través del cual se pudo obtener una visión del funcionamiento del sistema, siendo este la entrada fundamental para la realización del diseño.

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

3.1. Introducción

Para que un software se realice con éxito y el mismo tenga la calidad requerida se hace necesario emplear distintas métricas que permitirán erradicar problemas en los requisitos o casos de uso del sistema, lo cual hace que se logren los objetivos propuestos satisfactoriamente.

3.2. Método de validación de los resultados

En cualquier investigación es de gran importancia realizar el análisis de los resultados de la propuesta de solución al problema en cuestión, para de esta forma saber hasta qué punto se satisfacen las necesidades de los beneficiados. En la creación de un software también pasa lo mismo, se debe realizar la validación, la misma ayuda a saber al desarrollador si el sistema que están construyendo es lo que realmente espera el cliente. En el presente trabajo se utilizaron varios métodos para la validación de manera que fuesen validados los artefactos generados durante la investigación.

3.3. Aplicación de la métrica de la calidad de especificación de los requisitos

La métrica de la calidad de especificación de los requisitos mide la especificidad de los requisitos, permitiendo que los interesados puedan entenderlos de manera fácil y se puedan probar.

Primeramente se tiene **Nr** que representa el total de requisitos:

$$\mathbf{Nr} = \mathbf{Nf} + \mathbf{Nnf}$$

$$\mathbf{Nr} = 16 + 26$$

$$\mathbf{Nr} = 42$$

Donde **Nf** es el número de requisitos funcionales y **Nnf** es el número de requisitos no funcionales.

Luego de tener este dato se puede determinar la especificación (ausencia de ambigüedad) y para ello se plantea: $\mathbf{Q} = \mathbf{Nui} / \mathbf{Nr}$

Donde **Nui** representa el número de requisitos para los que los revisores tuvieron interpretaciones idénticas, es importante resaltar que cuando más cerca de 1 este el valor **Q**, menos ambigüedad presentara la especificación.

En el presente trabajo se realizaron dos revisiones para llegar a obtener un análisis sin ambigüedad en los requisitos, y para que se reflejara con más claridad las necesidades del cliente, en las cuales se obtuvieron los siguientes resultados:

Revisión 1: Durante la revisión de un total de **43** requisitos, entre funcionales y no funcionales, se detectaron de los **16** requisitos funcionales dos que no estaban claramente especificados, es el caso de Gestionar calendario de grabación y Mostrar calendario de grabación ya que los requisitos no expresaban con claridad su objetivo. Por lo que en todos los requisitos menos en 2, los revisores tuvieron interpretaciones bastante parecidas.

$$Q = Nui / Nr$$

$$Q = 41 / 42$$

$$Q = 0.97$$

Revisión 2: Durante esta revisión no se detectaron ambigüedades en los requisitos revisados, debido a la realización de un profundo análisis. Por lo que de los 43 requisitos con los que se cuentan los revisores tuvieron interpretaciones bastante parecidas.

$$Q = Nui / Nr$$

$$Q = 42 / 42$$

$$Q = 1$$

Luego de haber realizado la segunda revisión se llega a la conclusión de que los requisitos no presentan ya ningún nivel de ambigüedad, por lo que se verifica que estos cumplan con una serie de atributos o características necesarias para la especificación de requisitos.

Para que la especificación de requisitos sea reconocida de manera satisfactoria debe cumplirse que:

La Correctitud sea menor 0,10

La Completitud sea mayor que 0,90

La Consistencia sea menor que 0,20

Correctitud: La especificación de requisitos es correcta si todos los requisitos que figuran en ella reflejan alguna necesidad real. Esta métrica se propone determinar si la especificación de requisitos contiene todos los requisitos necesarios para satisfacer las necesidades del negocio y los interesados.

$X = \text{Correctitud}$

$D = \text{Total de observaciones de negocio realizadas en la validación}$

$T = \text{Total de requisitos revisados}$

$X = D/T$

$D = 1$

$T = 16$

$X = 1/16$

$X = 0,06$

Compleitud: La métrica pretende determinar si la especificación de requisitos es completa. Una especificación de requisitos está completa si los requisitos están relacionados a la funcionalidad, el desarrollo, las restricciones del diseño, los atributos y las interfaces externas.

$X = \text{Compleitud}$

$O = \text{Total de observaciones de formato}$

$S = \text{Total de secciones del documento}$

$X = 1 - O/S$

$O = 0$

$S = 0$

$X = 1 - (0/0)$

$X = 1$

Consistencia: La métrica se propone determinar si la especificación de requisitos es consistente. Una especificación de requisitos es consistente si ningún conjunto de requisitos descritos en ella son contradictorios o entran en conflicto.

X = Consistencia

C = Total de observaciones de consistencia

T = Total de requisitos revisados

$X = C/T$

C = 2

T = 16

X = 2/16

X = 0,12

Características necesarias para la especificación de requisitos:

Atributo	Descripción	Cumplimiento (%)
Correcta	La ERS es correcta si contiene todos los requisitos que el software debe satisfacer.	El 100% ERS son correctas.
No Ambigua	La ERS es no ambigua si cada requisito tiene una única interpretación.	El 100 % de las ERS son no ambiguas.
Completa	La ERS es completa si están incluidas todas las funciones requeridas por el cliente.	El 100 % de las ERS son completas.
Consistente	La ERS es consistente si ningún subconjunto de la misma entra en contradicción con otro subconjunto.	El 100% de las ERS son consistentes.

Clara	La ERS es clara si está escrita en un lenguaje apropiado.	El 100 % de las ERS son claras.
-------	-----------------------------------------------------------	---------------------------------

3.4. Matriz de trazabilidad

Consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo qué objetivos corresponden a cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos. La matriz de trazabilidad se usa también para ver si los casos de uso que tenemos satisfacen todos los requerimientos del sistema. Garantizando la integridad de los requisitos al implementar cambios en el sistema. (17)

Las matrices de trazabilidad pueden utilizarse para verificar y validar las especificaciones de un sistema, así como garantizar que en todos los documentos entregables finales estén incluidas las especificaciones del sistema. (17) A continuación se muestra una matriz de trazabilidad donde los casos de usos del sistema se encuentran ubicados de forma horizontal y todos los requisitos funcionales están alineados de forma vertical, de manera tal que corresponden entre ellos.

Matriz de trazabilidad del editor de configuración:

CUS RF	Gestionar servidor de grabación.	Manejar servidor de grabación.
RF15	X	
RF16		X

Matriz de trazabilidad del servidor de grabación:

RF \ CUS	Grabar.	Brindar información.	Cargar plugin de grabación.	Monitorizar.
RF1	X			
RF2			X	
RF3	X			
RF4				X
RF5		X		

Matriz de trazabilidad del cliente:

RF \ CUS	Gestionar calendario de grabación.	Grabar manualmente	Mostrar cámaras instaladas	Configurar cliente	Visualizar calendario de grabación
RF10			X		
RF11					X
RF12	X				
RF13		X			
RF14			X		

Matriz de trazabilidad del plugin de grabación:

RF \ CUS	Notificar eventos.	Grabar.	Monitorizar proceso de grabación.	Inicializar plugin.
RF6		X		X
RF7		X		X
RF8			X	
RF9	X			

3.5. Lista de chequeo

Son frecuentemente usadas en inspecciones o revisiones de artefactos generados en el proceso de producción de software; son listas de aspectos que deben ser completados o verificados. A continuación se representan dos tablas correspondientes a la lista de chequeo aplicada a los requisitos obtenidos.

Lista de chequeo utilizada en la Universidad de Las Ciencias Informáticas:

Evidencia	Nivel de importancia	Procedimiento	Respuesta
¿Existe(n) el(los) documento(s) de "Especificación de Requisitos"?	A	Revisión del documento	Sí
¿Se realiza el registro de control de versiones de los cambios del documento?	M	Revisión del documento	Sí
¿Se define el alcance del documento?	B	Revisión del documento	Sí

¿Se realiza una descripción de la funcionalidad de los requisitos en lenguaje natural?	M	Revisión del documento	Sí
¿Están descritos todos y cada uno de los requisitos?	A	Revisión del documento	Sí
¿Se identifican los requisitos que afectan la usabilidad, fiabilidad, eficiencia o soporte del sistema?	M	Revisión del documento	Sí
¿Están especificadas, si existen, todas las restricciones relacionadas con el diseño del sistema?	M	Revisión del documento	Sí
¿Están especificados los requisitos para la documentación y ayuda de los usuarios?	M	Revisión del documento	Sí
¿Se identifican los componentes que han sido adquiridos o comprados para incorporarlos al sistema?	M	Revisión del documento	Sí
¿Están definidas todas las interfaces que deberán ser soportadas por el sistema (Usuario, Hardware, Software, Comunicación)?	M	Revisión del documento	Sí
¿Están especificados todos los requisitos legales, tales como patentes, derechos de autor, marca comercial, logotipo entre otras así como cualquier requisito de licencia o restricción de uso?	M	Revisión del documento	Sí

¿Existe evidencia o referencias de las normas o estándares aplicables al sistema, por ejemplo estándares legales, de calidad, normas de usabilidad, etc.?	M	Revisión del documento	Sí
-----------------------------------------------------------------------------------------------------------------------------------------------------------	---	------------------------	----

3.6. Métricas de casos de uso del sistema

Para lograr la calidad del artefacto del caso de uso del sistema obtenido anteriormente se le aplicó un modelo de métricas. El modelo usa 4 atributos, los cuales son (completitud, comprensibilidad, concisión, no trivialidad).

Completitud: Un caso de uso es completo si especifica todo lo que deben hacer el actor y el sistema para alcanzar el objetivo del caso de uso. (18)

Comprensibilidad: Un caso de uso es comprensible si todos los tipos de lectores (cliente, usuario, jefe de proyecto) pueden entenderlo fácilmente con una mínima explicación del autor. (18)

Concisión: Un caso de uso es conciso si no incluye información innecesaria. (18)

No trivialidad: Un caso de uso es no trivial si su secuencia de pasos conduce al actor a conseguir el objetivo que persigue la realización del caso de uso. (18)

Para esta evaluación se le quita al 100% el 7% por cada error que posea. Cuanto más cerca este el resultado del 100% en cuanto a concisión, completitud, comprensibilidad y no trivialidad, mejor el resultado de las métricas. Una vez aplicada en dos revisiones las métricas pertinentes a los casos de uso del sistema, se obtuvo la siguiente información:

Primera revisión: Para el atributo concisión: Se encontró que los casos de uso Gestionar calendario de grabación, Monitorizar, Configurar cliente, Mostrar calendario de grabación, Configurar servidor de grabación y Grabar manualmente contenían información innecesaria por lo que surgió un error del 42%, lo que implica que el 58% de los casos de uso cumplen adecuadamente con el atributo.

Para el atributo comprensibilidad: Se encontró que los casos de uso Autenticar usuario y Mostrar cámaras instaladas contenían información que no permitían un fácil entendimiento, por lo que surgió un error del 14%, lo que implica que el 86% de los casos de uso cumplen adecuadamente con el atributo.

Segunda revisión: Luego de rectificadas los errores de la primera revisión, basándose en estos se efectuó una segunda revisión en la cual los casos de uso del sistema se encontraban en correcto estado en cuando a completitud, comprensibilidad, concisión y no trivialidad, por lo que se puede decir que los casos de uso cumplen en un 100% con el modelo de métricas.

3.7. Conclusiones parciales

En el presente capítulo se utilizaron métricas para la validación de los requisitos a través de las cuales se corrigieron los posibles errores, quedando todos los requisitos con la calidad requerida para su aprobación. Se puede concluir que el análisis obtenido posee una calidad aceptable, facilitando la continuación eficiente de la realización del sistema en etapas posteriores.

CONCLUSIONES GENERALES

Con la realización de la presente investigación, mediante el cumplimiento de las tareas y objetivos propuestos se llegó a las siguientes conclusiones:

- Los principales sistemas de grabación de video analizados, presentan la limitante de que dependen del hardware del fabricante, el cual incluye sus principales funcionalidades.
- El levantamiento de requisitos permitió obtener las funcionalidades requeridas, se realizó la documentación referente al análisis del sistema posibilitando el cumplimiento del objetivo trazado.
- Las métricas aplicadas para evaluar la calidad de los artefactos generados permitieron corregir todos los errores existentes para lograr como resultado la calidad requerida en los requisitos para su aprobación.
- El desarrollo del presente trabajo permitió la elaboración de todos los artefactos correspondientes al análisis del sistema, obteniendo como resultado una amplia y organizada documentación sobre los procesos que se desarrollan en el en el nuevo módulo de grabación del sistema de video vigilancia Suria.

RECOMENDACIONES

Tras haber cumplido con el objetivo trazado en esta investigación se recomienda:

- Utilizar los artefactos generados en el presente trabajo de diploma, como entrada a la realización del flujo de trabajo correspondiente al diseño.
- En próximas iteraciones o versiones del producto revisar los requisitos funcionales, con el objetivo de aumentar las funcionalidades del sistema.

REFERENCIAS BIBLIOGRÁFICAS

1. axis. axis.com.[En línea] [Citado el: 23 de noviembre de 2010.] http://www.axis.com/es/products/video/about_networkvideo/evolution.htm.
2. **Rolando Alfredo Hernández León, Sayda Coello González.** *El paradigma cuantitativo de la investigación científica.* 2002.
3. video digital. *video digital.* [En línea] [Citado el: 10 de diciembre de 2010.] es.wikipedia.org/wiki/Video_digital.
4. sistemas_de_grabacion_digital. *sistemas_de_grabacion_digital.* [En línea] [Citado el: 10 de diciembre de 2010.] http://www.lealsistemas.com.ar/cctv/sistemas_de_grabacion_digital.php.
5. Video Surveillance Guide. *Video Surveillance Guide.* [En línea] [Citado el: 15 de enero de 2011.] <http://www.video-surveillance-guide.com/history-of-video-surveillance.htm>.
6. Aventura technologies. *Aventura technologies.* [En línea] 2006. [Citado el: 12 de enero de 2011.] www.aventuratechnologies.com.
7. Voxdata. *Voxdata.* [En línea] [Citado el: 14 de enero de 2011.] <http://www.voxdata.com.ar/tiposdvr.html>.
8. Axis Communication. *Axis Communication.* [En línea] Axis. [Citado el: 3 de febrero de 2011.] <http://www.axis.com/products/video/software/index.htm>.
9. Panasonic. *Panasonic.* [En línea] [Citado el: 3 de febrero de 2011.] <http://www.panasonic.com>.
10. Scati labs SA. *Scati labs SA.* [En línea] [Citado el: 3 de febrero de 2011.] <http://www.scati.com/productos/detalle.php?id=4>.
11. Aula TI. Donde lo difícil de la Tecnología se hace fácil. *Aula TI. Donde lo difícil de la Tecnología se hace fácil.* [En línea] WordPress y Pixel. [Citado el: 3 de febrero de 2011.] <http://www.aulati.net/?p=405>.
12. UML. *UML.* [En línea] [Citado el: 5 de febrero de 2011.] <http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>.
13. **Sanchez, María A. Mendoza.** Metodologías De Desarrollo De Software. *Metodologías De Desarrollo De Software.* [En línea] [Citado el: 5 de febrero de 2011.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

14. **Evelyn Menéndez Alonso**. Monografias.com. *Monografias.com*. [En línea] [Citado el: 10 de febrero de 2011.] <http://www.monografias.com/trabajos73/herramientas-case-proceso-desarrollo-software/herramientas-case-proceso-desarrollo-software2.shtml>.
15. Enterprise Architect, Herramienta CASE. *Enterprise Architect, Herramienta CASE*. [En línea] [Citado el: 10 de febrero de 2011.] <http://www.sparxsystems.com.ar/products/ea.html>.
16. Modelo de dominio/Modelo conceptual. *Modelo de dominio/Modelo conceptual*. [En línea] [Citado el: 15 de febrero de 2011.] <http://bdigital.eafit.edu.co/bdigital/PROYECTO/P005.12CDT629/capitulo6.pdf>.
17. Entorno Virtual de Aprendizaje. *Entorno Virtual de Aprendizaje*. [En línea] [Citado el: 12 de abril de 2011.] <http://eva.uci.cu/mod/resource/view.php?inpopup=true&id=34917>.
18. **Pressman, Roger S**. *Ingeniería del Software. "Un enfoque práctico"*. 2005.

BIBLIOGRAFÍA

Rolando Alfredo Hernández León, Sayda Coello González. *El paradigma cuantitativo de la investigación científica.* 2002.

Sanchez, María A. Mendoza. Metodologías De Desarrollo De Software. *Metodologías De Desarrollo De Software.* [En línea] [Citado el: 5 de febrero de 2011.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

Evelyn Menéndez Alonso. Monografias.com. *Monografias.com.* [En línea] [Citado el: 10 de febrero de 2011.] <http://www.monografias.com/trabajos73/herramientas-case-proceso-desarrollo-software/herramientas-case-proceso-desarrollo-software2.shtml>.

Enterprise Architect, Herramienta CASE. *Enterprise Architect, Herramienta CASE.* [En línea] [Citado el: 10 de febrero de 2011.] <http://www.sparxsystems.com.ar/products/ea.html>.

Modelo de dominio/Modelo conceptual. *Modelo de dominio/Modelo conceptual.* [En línea] [Citado el: 15 de febrero de 2011.] <http://bdigital.eafit.edu.co/bdigital/PROYECTO/P005.12CDT629/capitulo6.pdf>.

Entorno Virtual de Aprendizaje. *Entorno Virtual de Aprendizaje.* [En línea] [Citado el: 12 de abril de 2011.] <http://eva.uci.cu/mod/resource/view.php?inpopup=true&id=34917>.

Pressman, Roger S. *Ingeniería del Software. "Un enfoque práctico".* 2005.