“Universidad de las Ciencias Informáticas”

Facultad 6

Título: Gestor. Módulo para el Sistema de Video Vigilancia Suria.

Trabajo de Diploma para optar por el título de

Ingeniero de Ciencias Informáticas

Autor: Axel Rodríguez Durañona.

Tutor: Edmis Deivis Semanat Aldana

Ciudad de la Habana, 2012

DECLARACIÓN DE LA AUTORÍA:

Declaro que soy el único autor del presente 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 ____ días del mes de _____ del año 2012.

Autor:

_____

Axel Rodríguez Durañona

Tutor:

_____

Edmis Deivis Semanat Aldana.

DEDICATORIA:

A mi madre por todo su amor y cariño aunque no pueda estar conmigo ni verme en este momento.

A mi papá por todo su cariño, afecto incondicional y sus enseñanzas que me han ayudado para la vida.

A mis amigos y colegas que han estado junto a mí a todo lo largo de todos estos años.

Agradecimientos:

A mis padres.

A mi Familia por todo el amor, el afecto y el apoyo que me han profesado todos estos años.

A todos Profesores con que he podido interactuar a lo largo de todo este tiempo y que me han ayudado a formarme tanto docentemente como en lo referente a todos los valores humano necesarios para la vida.

A mis amigos.

RESUMEN:

Actualmente existen diferentes sistemas de Video -Vigilancia estos ayudan a la seguridad y vigilancia de cualquier institución o empresa que lo requiera. El auge de las nuevas tecnologías de la información ha posibilitado que una gran cantidad de empresas y organizaciones posean un sistema de seguridad que realice la captura, seguimiento de videos, catalogación, y el manejo de los datos en dichos sistemas.

Dentro de los sistemas de Video Vigilancia se puede intercomunicar todos los módulos entre sí directamente, pero esto puede traer problemas, por ejemplo si se adiciona un nuevo módulo los demás deben ser modificados uno a uno para que puedan comunicarse y cooperar con dicho módulo, lo cual trae como consecuencia que para una gestión más organizada se incluya dentro del sistema al módulo gestor para comunicar a todas las partes entre sí. Es por ello que este trabajo se centra en el objetivo de realizar un sistema de gestión en el que se incluyen la manipulación de información y eventos del proyecto.

El presente trabajo propone desarrollar un software que facilite el manejo y flujo de la información de un sistema de video y vigilancia. Así se hará más fácil el uso y chequeo de la información obtenida por las cámaras de seguridad en cualquier institución de nuestro país que lo requiera. Esto posibilitará que la información del sistema se gestione de manera correcta en donde se haga uso de dicho sistema de Video-Vigilancia.

Palabras Claves: cámaras, gestión, sistema, seguridad, video.

Índice

INTRODUCCIÓN:

Desde el surgimiento de la humanidad, el hombre siempre ha tenido la meta de proteger sus bienes materiales mediante la vigilancia esto ha traído como consecuencia que se hayan implementado sistemas de distintos tipos para lograr esta meta de la mejor manera. Con el siglo 20 y el desarrollo de la televisión y las telecomunicaciones los sistemas de vigilancia han evolucionado a un punto más alto donde el factor humano ya no juega un papel tan relevante como en tiempos antiguos donde el guardia era el único encargado de vigilar dichos bienes.

En nuestro país se realiza gran cantidad de inversiones en tecnología de punta para todas las instituciones que lo necesiten. De ahí que se necesite proteger todos esos bienes que han sido obtenidos con el sudor de nuestro pueblo de posibles agentes delictivos que amenazan la seguridad de los mismos, por lo que la seguridad de dichos bienes de manera general tenga gran importancia en la actualidad, para lo cual se implementan políticas y mecanismos para su cuidado.

Todos los sistemas de vigilancia manipulan una cantidad considerable de información lo cual trae como consecuencia que dicha información debe ser manipulada con sumo cuidado. Con la introducción de las nuevas tecnologías se han ido perfeccionando la gestión de la información y se han automatizado funciones que en tiempos de antaño solo era posible hacerlas de manera manual, esto ha posibilitado la apertura de nuevos y amplios horizontes para el desarrollo de los sistemas de video-vigilancia.

El proyecto Video Vigilancia Inteligente se encuentra enfocado en el proceso de captura, detección de objetos, y seguimiento de objetos, para de esta forma lograr llevar a cabo la implementación de un sistema que permita garantizar la seguridad de las empresas y centros que lo requieran en distintos lugares de nuestro país.

Dentro de la estructura de este proyecto se incluyen un conjunto de cámaras cuyas grabaciones pasan por una serie de módulos para ser procesadas y analizadas de manera integral y minuciosa, y que dichos datos persistentes (dígase grabaciones y fotos) puedan ser gestionados dentro de un sistema de procesamiento de datos asegurando la grabación y visualización de estos vídeos obtenidos por cada cámara del sistema. Se necesita de un nuevo módulo que permita la unión de todos los (módulos) del sistema el grabador, el visor y el recuperador, el cual debe coordinar la comunicación de cada una de las partes del sistema logrando un correcto funcionamiento del mismo.

Con la intención de extender el desarrollo de las nuevas tecnologías hacia todos los sectores y lugares de la sociedad, el país se encuentra inmerso en un proceso de informatización. Las empresas cubanas van ajustándose al desarrollo informático vertiginoso y el campo de la Video-Vigilancia no escapa a ello.

La tecnología de la gran mayoría de dichas empresas se encuentran afrontando un proceso de cambio y evolución, por lo que se hace necesario que dicho cambio se realice y dando lugar a políticas que favorezcan el uso de las TIC en el sector que nos ocupa (La Video-Vigilancia) de manera consciente y eficaz. Esto conducirá que la empresa y la sociedad realicen una verdadera revolución tecnológica. La Situación planteada anteriormente conduce al siguiente **problema a resolver:**
¿Cómo intercomunicar los módulos del Sistema de Video Vigilancia SURIA y centralizar la gestión de los datos en el sistema?

Quedando identificado de esta manera el problema, para dar solución al mismo se enmarca el **objeto de estudio** en los procesos de gestión de datos y coordinación de tareas.

El **campo de acción** queda definido en la gestión y estudio del componente gestor, en su variante Tablero de Control para el Sistema de Video Vigilancia SURIA, así como los demás elementos involucrados que puedan surgir y estén vinculados ha dicho componente.

Para poder desarrollar y dar una respuesta satisfactoria al problema anteriormente planteado se plantea como **objetivo general:**

Desarrollar un sistema Gestor para el Sistema de Video Vigilancia que permita gestionar todos los datos persistentes del proyecto.

Para poder dirigir y guiar la investigación se plantea la siguiente **hipótesis**: con el desarrollo de un módulo Gestor para el Sistema de Video Vigilancia SURIA se logrará un aumento de calidad en cuanto a la coordinación, eficacia, rapidez con que se comunican los diferentes módulos (Grabador, recuperador, visor etc.).Logrando de esta manera el desarrollo de un producto final de mayor validez que contribuirá a proveer un servicio que actualmente no existe y se necesita dentro del proyecto.

Para desarrollar la investigación y lograr los objetivos planteados se plantean las siguientes tareas**:**

1-Realización de un estudio y análisis de los diferentes sistemas de gestión que existen para la el manejo de flujos de video y fotos obtenidas mediante cámaras ip.
2-Selección de plataforma más correcta para el desarrollo de la aplicación.
3-Realización de un estudio que conlleve a identificar los requerimientos funcionales y no funcionales del mismo.
4-Desarrollo y modelado de los casos de usos del sistema.
5-Implementación de todas las funcionalidades del módulo Gestor.
Todo esto llevado a cabo para obtener los siguientes **resultados esperados:**

- Documentos ingenieriles asociados a la investigación.
- Documento de Tesis.
- Una versión en QT del Módulo Gestor existente en el Sistema de Video Vigilancia SURIA.

Métodos teóricos:

Analítico-Sintético*:*

Este método es el que nos ha permitido poder realizar una profunda investigación de toda la documentación referente al objetivo que nos ocupa, de esta manera se pudo obtener cada elemento que se encuentra vinculado al objeto de estudio, y además nos trazó a partir del análisis bibliográfico el camino a seguir para el desarrollo e implementación del sistema.

Análisis histórico-lógico*:*

Mediante el uso de este método hemos podido analizar y llegar a comprender cuál ha sido la evolución y manejo de los sistemas gestores, esto ha servido para entender y mejorar la esencia del objeto de estudio.

Métodos empíricos:

Observación*:* Este método nos ha sido muy útil, ya que todo lo que crearemos será visualizado a la par que se implementa para observar los cambios y modificaciones que ocurran en el problema y la aplicación que nos ocupan.

Este método empírico (observación) es esencial en la investigación científica. Ya que nos permite conocer de manera directa de los objetos y fenómenos con que interactúa el objeto de estudio, a partir de objetivos previamente existentes.

El documento está estructurado de la siguiente manera:

El **Capítulo 1:** hace referencia y descripción de los sistemas gestores de manera general, así como estado del arte de los gestores existentes en la actualidad que son los más usados y poseen semejanza con el que se va a implementar. Y por último, se hace un profundo análisis de las tecnologías, metodologías y herramientas que serán usadas en el desarrollo del sistema.

El **Capítulo 2:** se expone las principales características del sistema. Además, hace referencia el modelo de dominio de la aplicación se levantaran los requerimientos propios del sistema, tanto funcionales como no funcionales, y el modelo de casos de usos del sistema.

El **Capítulo 3:** muestra los resultados que se han obtenido en el desarrollo de los procesos de análisis y diseño del sistema. Se desarrollarán los artefactos pertenecientes a los flujos de trabajo análisis y diseño, dígase diagramas de clases y diagramas de interacción que fueron necesarios para obtener una mayor claridad a la hora de elaborar el módulo que se propone.

Y, en el **Capítulo 4:** se da a conocer el modelo de implementación que se obtuvo como resultado del análisis y diseño estando compuesto por su correspondiente diagrama de despliegue, y su diagrama de componentes.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

INTRODUCCIÓN*:*

En este capítulo se abordan conceptos relacionados con los sistemas de video vigilancia. Se hace referencia al estado del arte en que se encuentran en el mundo los sistemas relacionados con la gestión en la Video-Vigilancia presentándose algunas soluciones sobre los mismos y se abordan todas las características relacionadas con la evolución de dichos sistemas, además se hace un análisis sobre las principales tecnologías, metodologías y herramientas a ser usadas en el desarrollo de este trabajo.

1.1-CONCEPTOS ASOCIADOS AL DOMINIO DEL SISTEMA

A continuación se describen los principales conceptos relacionados con la concepción del sistema, de esta forma quedan claros algunas definiciones que servirán como punto de partida para el desarrollo del presente trabajo, cada concepto sirve como punto de partida para el siguiente ya que lo describe y complementa.

Gestión:

Aquí queda claro un concepto fundamental de este sistema, el mismo describe la definición de una palabra que va a ser repetida en múltiples ocasiones a lo largo de este trabajo y que servirá como punto de inicio para la implementación del este gestor. En dicho concepto quedan claro todas las características fundamentales que se espera de la gestión.

El término gestión, por lo tanto, es una serie de acciones que se llevan a cabo para dar solución a un asunto o concretar un proyecto en específico. La gestión es también la forma de llevar a cabo la administración y manejo de una institución, empresa, negocio, sistema o trabajo (1).

Sistema:

Este concepto es esencial e idóneo para este trabajo, ya que describe de manera concreta y específica que es lo que se espera de un sistema de cualquier tipo (en este caso va ser un sistema de gestión de Video-Vigilancia) así como describe las principales características y propiedades con que debe contar el mismo.

Un sistema real, es una entidad material formada por componentes organizados que interactúan de forma en que las cualidades del conjunto no pueden deducirse por completo de las propiedades de la partes (denominadas propiedades emergentes).Un sistema ejecuta una función imposible de realizar por cualquiera de las partes individuales (1)**.**

Sistema de Gestión:

A continuación se describe un concepto que es esencial para este trabajo ya que evidencia la unión de los dos conceptos fundamentales en este documento **Sistema** y **gestión** y muestra cómo los mismos se complementan en uno solo describiendo sus características fundamentales. A continuación se define el mismo.

El conjunto de sistemáticas o metodologías orientadas a un mismo propósito, junto a los recursos, procesos, actividades y responsabilidades asociadas a las mismas, conforman un Sistema de Gestión (2).

Los Sistemas de Gestión se traducen en acciones y procedimientos planificados y organizados por medio de los cuales se busca conseguir unos resultados específicos. Cuando una empresa implanta un sistema de este tipo solo lo hace para gestionar un aspecto puntual o múltiples procesos de negocio, como por ejemplo la calidad de sus productos y servicios (3).

Cámaras IP:

Aquí podemos apreciar un nuevo elemento esencial para todos los **sistemas de gestión** de video-vigilancia ya que las cámaras ip constituyen la base de la entrada de datos dichos sistemas, ellas debido a su eficiencia, se pueden utilizar en una PC o un servidor para de ésta manera poder llevar a cabo la visualización, grabación y recuperación de videos e imágenes. Una de las mayores ventajas de estas cámaras es su capacidad para la visión del vídeo y sonido en tiempo real con el gran beneficio de que se pueden acceder a ellas a través de internet, es decir se tiene acceso desde cualquier lugar y en cualquier momento a los datos obtenidos por dichas cámaras.

De esta manera se define que: las cámaras IP son dispositivos que permiten capturar y emitir flujos de videos en una red de comunicación (ejemplo Internet o de área local) usando la tecnología TCP/IP sin necesidad de una computadora. Dentro de las Cámaras IP, además de la cámara en sí, tiene incorporado en su estructura un microordenador que es lo que permite cumplir con las funciones anteriormente descritas de comunicación desde la red (3).

Las Cámaras IP son un una moderna tecnología de seguridad y vigilancia. Las cámaras y los servicios se pueden utilizar con redes IP ya instaladas, de ésta forma se elimina la necesidad de implementar un nuevo sistema de cableado dedicado.

Videos:

Y por último y no menos importante se describe el concepto de videos ya que ellos son captados u obtenidos precisamente por medio de las **cámaras ip** que se encuentren instaladas en el sistema, es decir las cámaras son el medio fundamental de visualizar y capturar los videos en los proyectos de video-vigilancia.

Concepto de Videos: Está compuesto por fotogramas, que son cada una de las imágenes individuales de una secuencia o animación que unidos son las escenas, representan un conjunto de tomas relacionadas ente si, los videos son captados por medios electrónicos digitales o analógicos. En la actualidad, el término hace referencia a distintos formatos como VHS y Betamax, también se incluyen los formatos digitales, como:MPEG1, MPEG2, MPEG4, H264, RMVB. La calidad del video estará determinada por distintos factores, como el método de captura y el tipo de almacenamiento elegido (4).

.

1.2- ESTADO DEL ARTE DE LOS SISTEMAS DE VIDEO Y VIGILANCIA A NIVEL MUNDIAL.

Existen múltiples sistemas de video y vigilancia en la actualidad que sirven para el manejo y gestión de videos e imágenes obtenidos a partir de las cámaras IP y su mercado lo constituyen cualquier tipo de empresa, compañía o institución nivel nacional o mundial. Esta evolución fue partiendo de las necesidades existentes de poder vigilar bienes materiales o de otra índole en donde se requiera, en cada sistema debe existir una manera de poder manipular dichos datos para prever la ocurrencia de errores o perdidas en los mismos, y además se hace necesario que para manipular estos datos se cumplan con una serie de características que definirán la gestión de la información como, fiable, de fácil uso y manejo, y que ella no se realice de manera engorrosa ni compleja para usuarios que no posean un gran conocimiento de la tecnología.

La gestión un sistema constituye el núcleo del sistema ya que mediante ella se controla la comunicación entre cada módulo del mismo, y hace que estos interactúen entre sí a la hora de realizar la búsqueda de la información que necesiten y no que cada uno por su lado tenga que buscarla dentro de los otros módulos, ya que si se agregan nuevas partes al sistema todos ellos tendrán que ser modificados individualmente para poder trabajar con los otros que sean agregados.

La evolución de los sistemas de video y vigilancia ha ido a la par que se desarrollan las nuevas tecnologías. Por último la gestión en los sistemas de

Video-Vigilancia debe ser considerada como un requisito indispensable y no debe ser ignorada ni minimizada su importancia ya que ella se centra en encontrar solución al manejo de la información de dichos sistemas que en muchas ocasiones es extremadamente extensa.

Las posibilidades de los sistemas de video vigilancia, dependerán del nivel de la seguridad que se necesite en dependencia del lugar donde sea implementado. Como en toda tecnología hay sectores son los más interesados para la adopción e incorporación de dichos sistemas. Ejemplos de ello son: administraciones públicas (protección de edificios públicos, para el control del tráfico, acciones delictivas y prevención de actos terroristas), el comercio, los bancos, zonas turísticas, tiendas, lugares para el seguimiento de la naturaleza y los animales, así como otros.

A continuación se relacionan algunos Sistemas de Video-Vigilancia, todos ellos compuestos por diferentes módulos y funcionalidades, esto se debe a la evolución y avance de los mismos en todos estos años.

AXIS Camera Station: Desarrollado por Axis, empresa líder en la producción sistemas de video-vigilancia, que ofrece toda una serie de ventajas y funcionalidades. Entre las ventajas se incluyen la accesibilidad remota (gracias al uso de cámaras IP), la alta calidad de imagen, la gestión de eventos y las capacidades de vídeo inteligente, así como las posibilidades de una integración sencilla y una escalabilidad, flexibilidad y rentabilidad mejoradas.

El AXIS Camera Station por tanto es un completo sistema de supervisión, grabación y gestión. Diseñado para los productos de vídeo en red de Axis, este software de vigilancia IP que funciona con cámaras de red y codificadores de vídeo de Axis, proporciona funciones de visualización de vídeo, grabación y gestión de eventos. El software posee numerosas funciones de búsqueda de eventos y videos grabados así como su manejo y gestión (5).

Otras características:

- Visualización y control remoto de uno o varios lugares de emplazamiento.
- Diseñado para sistemas pequeños y medianos.

- Manejo sencillo y de fácil uso.
- Permite ver, grabar y exportar vídeo de alta calidad en los formatos de compresión H.264, MPEG-4 y Motion JPEG.
- Puede ser configurado para visualización en directo, incluyendo un mapa del sitio para una visión más completa y extensa.

VideoSphere:

Este sistema de la compañía March Networks Corporation es un software de alto rendimiento que constituye la piedra angular de la gama de la gestión de vídeo inteligente, VideoSphere Gestión, el sistema de gestión es ideal para aplicaciones de monitoreo promedio a gran escala. Permite a las empresas transmitir vídeo de alta calidad obtenido de 128 cámaras, y el almacenamiento centralizado. Múltiples servidores y alojamiento de almacenamiento que pueden ser agrupados para instalaciones de apoyo, incluidos miles de cámaras. El Sistema de Gestión de VMS utiliza tanto la compresión de vídeo H.264 MPEG-4 para asegurar una alta eficiencia de uso de ancho de banda de red y capacidad de almacenamiento. Esto elimina la necesidad de video por separado y ayuda reducir el costo total de propiedad a través de convergencia de recursos en la infraestructura de gestión (6).

Software de control: (Video SphereSite Manager**):**

Video SphereSite Manager es la aplicación software de escritorio para configurar y explotar el sistema de gestión de los recursos desde cualquier punto de la red negocio. Permite a los administradores pronta puesta en marcha de cada servidor después de instalar el software, e incluye una función de detección automática que reconoce activos y asigna direcciones IP y espacio de almacenamiento para todas las cámaras de vídeo Videosphere y todos los codificadores en la red.

VALORACIÓN DEL ESTADO DEL ARTE:

Analizando en profundidad todos los sistemas anteriormente vistos. Se evidencia que son propietarios y que se necesita la contratación de servicios privatizados para el uso, aprovechamiento y puesta en práctica de cualquiera de estos sistemas. Partiendo desde la base que el sistema que se pretende desarrollar es un software que no es privativo se puede decir que: a pesar de

que existen mundialmente varios y reconocido sistemas para la Video-Vigilancia que incluyen dentro de sí todo lo necesario para el seguimiento de videos, reconocimiento de patrones e identificación de sujetos específicos, los mismos no satisfacen las necesidades del proyecto por lo que se hace necesario desarrollar uno diferente, esto se debe en gran medida a que la mayoría de estos sistemas al ser propietarios solos son compatibles con las tecnologías(cámaras IP, software y pizarras etc.) que son vendidas por la empresa que ofrece el sistema por lo que se hace muy costoso la aplicación del mismo así como su mantenimiento de ahí que se debe desarrollar el sistema que se desea dentro del proyecto de Video-Vigilancia inteligente.

Además en cuanto a razones técnicas y prácticas, prácticamente ninguna de estas aplicaciones brindan soluciones para gestionar recursos de un sistema de Video-Vigilancia de la forma más efectiva pues estas dependen principalmente la compra de todo el equipamiento necesario para el funcionamiento del mismo el cual es costoso y de muy difícil mantenimiento y manejo ya que para ello se necesita en ocasiones capacitación en dicha tecnología lo cual también es costoso por lo que no es factible contratar estos servicios. A manera de resumen podemos concluir que Los gestores de Video-Vigilancia son sistemas propietarios a los cuales hay que solicitar su compra y pagar por los mismos haciéndose muy difícil su adquisición y manejo.

1.3-TECNOLOGÍAS Y HERRAMIENTAS A USAR EN EL DESARROLLO DEL SISTEMA.

Las tecnologías que se usarán para el desarrollo del sistema se encuentran en relación con el lenguaje de programación seleccionado en dependencia de las necesidades de implementación y desarrollo del sistema:

Lenguaje de programación C++:

Características del C++:

1-El C++ es un lenguaje orientado a objetos el cual posee características y cualidades de las que carecía el lenguaje C con lo cual fue enriquecido y mejorado (7).

2-El C++ es uno de los lenguajes más potentes porque permite programar a alto y a bajo nivel, es complicado porque se debe hacer casi todo de manera manual.

3-El C++ es una extensión del lenguaje C. Este lenguaje es un lenguaje de programación mixto, se le puede compilar. Una de las ventajas que ofrece es que es mucho más sencillo de aprender para los programadores que ya conocen el lenguaje C.

4-Tiene una enorme compatibilidad con el C principalmente por dos razones: Por la gran cantidad de código C que comparten, y para facilitar el paso de los programadores de C al nuevo lenguaje C++.

El C++ por lo tanto es un lenguaje muy completo, el mismo permite la programación multiparadigma, y es a su vez conciso, eficaz y claro. Desde su surgimiento fue el lenguaje utilizado por la mayoría de los desarrolladores para sus proyectos. Por esta razón este leguaje es ideal para el desarrollo de cualquier aplicación en general.

Microsoft Visual C# .net: es un lenguaje de programación creado para realizar algunas de las aplicaciones que se ejecutan en .NET Framework. C# el lenguaje elite de Microsoft, es simple, eficaz, de fácil uso y documentación y orientado a objetos. Posee una gran gama de innovaciones así como un completamiento de código muy completo lo cual hace que usuarios que no sean muy avezados en la materia de programación puedan usarlo de manera sencilla y directa. Este lenguaje es el lenguaje paradigma de la plataforma de desarrollo.net (8).

C# es prácticamente la máxima evolución a que se puede llevar un leguaje para facilitar el trabajo de diseño e implementación de algún sistema, él es ideal para desarrollar cualquier tipo de software y además permite un

seguimiento y detección de errores excelente, no por gusto es uno de los productos más solicitados de Microsoft.

Herramientas (IDE de desarrollo):

Los IDE se deben seleccionar en dependencia de los lenguajes en que se piensa desarrollar, puesto que no todos los IDE soportan estos lenguajes candidatos.

El **QTCreator** utiliza el lenguaje C ++, pero hace un amplio uso de un generador de código especial (llamado el *Meta ObjectCompiler,* o *MOC),* junto con varias macros para enriquecer el lenguaje. Qt también se puede utilizar en varios otros lenguajes de programación a través de enlaces de lenguaje.

Algunas de sus características son:

- Compatibilidad multiplataforma con un sólo código fuente.
- Performance de C++.
- Disponibilidad del código fuente.
- Excelente documentación.

Aquí se evidencia un IDE de desarrollo que será usado para este sistema debido a su potencia en el desarrollo de aplicaciones de software, el mismo es perfecto debido a la utilización del c++ como lenguaje de programación porque el mismo nos permite realizar prácticamente cualquier funcionalidad que se requiera por parte de un proyecto específico (9).

Ingeniería y modelado UML:

Visual Paradigm:

Las Herramientas CASE para el modelado UML son aplicaciones informáticas usadas por analistas, ingenieros de software y desarrolladores destinados a aumentar la productividad en el desarrollo de software. Entre ellas una de las más destacadas es Visual Paradigm. Esta es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

Es además una herramienta de modelado UML muy profesional el mismo soporta el ciclo de vida completa del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

Visual Paradigm por otro lado brinda la posibilidad de generar código a partir del modelo de clases del diseño y permite realizar ingeniería inversa para plataformas como: .Net, Java y PHP. Posibilita la integración con otras herramientas de desarrollo, como son Visual Studio y Eclipse, BorlandJBuilder, NetBeans, IntelliJ IDEA, JDeveloper lo que facilita el trabajo de los desarrolladores, puesto que pueden modelar y programar en la misma plataforma. VP también permite una integración con los elementos del paquete Office de Microsoft como el Excel, el Word y el Power Point, los diagramas generados se pueden modificar directamente desde los documentos, sin la preocupación de perder el original, ya que estos se encuentran embebidos dentro del documento Office.

Enterprise Architect:

Provee un ambiente de modelado para trabajo y modelado UML que abarca el ciclo de vida completo del desarrollo de software, con herramientas que pueden proveerle una estructura completa en modelado de negocio, diseño de software, ingeniería de sistemas, arquitectura de UML, gestión de requisitos, testing y mucho más. Una herramienta que abarca el ciclo de vida completo.

EA le permite tanto **modelar en UML 2.0**, como rastrear la información importante de proyecto con artefactos de diseño. Por ejemplo, guardando la información de prueba, el riesgo, la métrica, cuestiones, defectos, cambios, archivos y otros directamente mediante los elementos del modelo que a que hacen referencia para ayudar a manejar proyectos grandes y complejos (10).

EA soporta la **generación y la ingeniería reversa de código fuente** para muchos lenguajes populares, incluyendo C++, C#, Java, Delphi, VB.Net, Visual Basic y PHP(16).

El Lenguaje de Modelamiento Unificado proporciona ventajas significativas, ayuda a construir complejos y rastreables modelos de sistemas de software en forma constante. Enterprise Architect soporta este proceso en un ambiente fácil de usar, rápido y flexible.

Por todo lo antes descrito es que consideramos que esta herramienta de modelado UML® Enterprise Architect es perfecta para este ciclo de desarrollo, porque además de ser herramienta muy potente y flexible para la plataforma de Windows soporta el análisis de negocio, provee un ambiente amigable óptimo para el desarrollo de modelado UML, la administración de proyectos, de requerimientos funcionales y de diseño del negocio.

Desarrollo de documentos: Microsoft Office 2007 es una versión de la suite ofimática Microsoft Office de Microsoft y sucesora de Microsoft Office 2003. Originalmente conocido como ***Office 12*** durante su ciclo beta, fue lanzado el 30 de noviembre de 2006 al mercado empresarial y el 30 de enero de 2007 al público en general, coincidiendo con el lanzamiento oficial de Windows Vista. Office 2007 incluye nuevas características, la más notable es la nueva interfaz gráfica llamada *Office Fluent*, también conocido como "interfaz de cinta" (en inglés "*ribbon*"),que reemplaza al menú y a la barra de herramientas que fueron características desde su inicio.

Sistema gestor de Bases de datos:

PostgreSQL 8: La versión 8.3, ha continuado con el rápido desarrollo de esta base de datos de código abierto. Esta versión contiene una gran cantidad de mejoras para que la administración, consulta y programación en PostgreSQL sea más fácil que nunca. Con las 293 funcionalidades nuevas o mejoradas en la versión 8.4.La versión 8.4 hace el análisis de datos mucho más sencillo a través de funcionalidades avanzadas de ANSI SQL: 2003, como las funciones Windows, expresiones comunes de tabla y joins recursivos (11).

MySQL: Es el sistema de gestión de bases de datos SQL Open Source más popular, lo desarrolla, distribuye y soporta MySQL AB. MySQL AB es una compañía comercial, fundada por los desarrolladores de MySQL. Es una compañía Open Source de segunda generación que une los valores y metodología Open Source con un exitoso modelo de negocio.

1.4 –METODOLOGÍAS A USAR EN EL DESARROLLO DEL SOFTWARE DEL SISTEMA:

Las metodologías ingenieriles han estado presentes durante mucho tiempo en todos los sistemas, la crítica más frecuente a estas diferentes metodologías es que son burocráticas y muy pesadas en ocasiones. Se deben realizar tantos pasos para seguir una metodología que el ritmo entero del desarrollo del sistema se retarda en ocasiones, aunque las mismas son un eslabón inviolable e indispensable en el proceso de desarrollo de software. Generalmente las metodologías de desarrollo de software se dividen en dos grandes grupos caracterizados debido al trabajo realizado en toda documentación del desarrollo del sistema, estos son Metodologías Ágiles de Desarrollo y Metodologías Tradicionales o Robustas.

Metodología Rup.

El proceso unificado de desarrollo (RUP) es una metodología para la ingeniería de software que va más allá del mero análisis y diseño orientado a objetos para proporcionar una familia de técnicas que soportan el ciclo completo de desarrollo de software. El resultado es un proceso basado en componentes, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental (12).

Características principales de RUP:

1.-Guiado por los Casos de Uso: Los Casos de Uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba.

2.-Centrado en la arquitectura: Los modelos son proyecciones del análisis y el diseño constituye la arquitectura del producto a desarrollar.

3.-Iterativo e incremental: Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo.

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. Se tiene una arquitectura más robusta en las fases finales del proyecto.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en los distintas actividades. Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una línea base de la arquitectura.

Durante la fase de Inicio las iteraciones ponen mayor énfasis en actividades modelado del negocio y de requisitos.

En la fase de Elaboración, las iteraciones se orientan al desarrollo de la línea base de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la línea base de la arquitectura. En la fase de Construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones. Para cada iteración se selecciona algunos Casos de Uso, se refina su análisis y diseño y se procede a su implementación y pruebas. En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios (13).

Metodología XP:

La **programación extrema** o *eXtremeProgramming* (XP) es un enfoque de la ingeniería de software que define procesos ágiles de desarrollo de software. Al

igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los objetivos de XP son muy simples: la satisfacción del cliente. Esta metodología trata de dar al cliente el software que él necesita y cuando lo necesita. Por tanto, debemos responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final de ciclo de la programación. El segundo objetivo es potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software. XP define cuatro variables para proyectos de software: coste, tiempo, calidad y ámbito.
Sus características fundamentales son:

1-Desarrollo Iterativo e incremental.

2-Simplicidad.

3-Refactorización del código.

4-Pruebas unitarias conjuntas.

5-Integración del equipo de programación con el cliente.

6-Correción de errores.

7-Programación en parejas.

Por todo lo antes descrito se puede arribar a la conclusión de considerar la programación extrema como una de las mejores metodologías para lograr el desarrollo de un sistema, esta metodología puede ser aplicada de manera dinámica durante el ciclo de vida del software.

1.5–CONCLUSIONES:

En este capítulo se ha obtenido una visión más clara de los diferentes conceptos con relación a la gestión de Video-Vigilancia, se analizaron las características esenciales que se tendrán en cuenta a la lo largo de este trabajo. Aquí se obtuvo como resultado de este capítulo que: se ha seleccionado RUP como metodología para el desarrollo de este sistema porque es mundialmente reconocida como una de las eficaces y seguras a la hora de llevar a cabo la implementación de un sistema orientado a objetos, además RUP provee una estructura clara en relación a que se debe hacer en cada momento del desarrollo.

Se usará el Enterprise Architect para el modelado UML ya que el mismo es ideal porque que soporta el ciclo de vida completo del desarrollo de software orientados a objetos, construcción, pruebas y despliegue por lo que es idóneo para el sistema, y como lenguaje de programación el C++ , sin ser seleccionado el C# de Microsoft, que es sin dudas, más sencillo y de fácil entendimiento porque C++ es el que se usa fundamentalmente para desarrollar aplicaciones y soluciones basadas en el Framework QTCreator y porque este es un lenguaje de programación muy potente que provee soluciones para los problemas que puedan surgir a lo lago de la implementación de cualquier tipo de aplicación.

Además en el contexto de este capítulo se arrojaron nuevos conocimientos acerca de algunas de las principales soluciones que existen en el mundo para la gestión y manejo de los videos e imágenes analizando algunas de sus características fundamentales que las definen.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

INTRODUCCIÓN.

Este capítulo define los elementos que intervienen en la solución del problema y la relación entre ellos. En el marco del mismo se dan a conocer y se describen conceptos claves que compondrán este sistema, por otra parte se muestra la estructura del Modelo de Dominio es decir se comienza el modelado del sistema. Además, se referencian los requerimientos funcionales y no funcionales, definiendo y agrupando los primeros como Casos de Uso del Sistema, con el fin de estructurar el Diagrama de Casos de Uso del Sistema y también se muestran los actores del sistema, así como se lleva a cabo la descripción de cada caso de uso del sistema de manera expandida, exponiendo cada uno de ellos por su ciclo de desarrollo.

2.1-DISEÑO DEL MODELO DE DOMINIO.

En este trabajo en específico el modelo del dominio hace hincapié en la manera en el sistema va a manejar la información y obtenerla a partir de su unión al gestor.

El modelo de dominio es un diagrama de clases en el que se muestra:
1-Conceptos u objetos asociados al dominio del problema o sistema: llamados clases conceptuales.

2- Asociaciones y/o relaciones entre las clases conceptuales.

3-Atributos de la clase conceptuales.

2.1.1-CONCEPTOS FUNDAMENTALES DEL SISTEMA:

Zona: Un área determinada o espacio de un lugar en específico.

Grupos: Se define grupo como el conjunto de dos o más individuos que se relacionan y son interdependientes y que además posean características comunes que los identifiquen como de un mismo tipo de alguna manera.
Operador: El operador es el encargado de mediar funcionalmente entre una tecnología o dispositivo tecnológico y otra instancia que puede ser, otro operador u otra instancia de un sistema o el público general.

Administrador del sistema: Son individuos que en una organización y sistema que dirigen las actividades de otros así como su control. Estos también poseen responsabilidades operativas que se manifiestan en el manejo y gestión de los distintos componentes que componen el mismo.

Gestión: Es la acción o efecto de gestionar o administrar, es decir realizar diligencias que conduzcan al logro de un objetivo cualquiera. El gestor es el encargado de dirigir, ordenar, disponer u organizar un sistema.

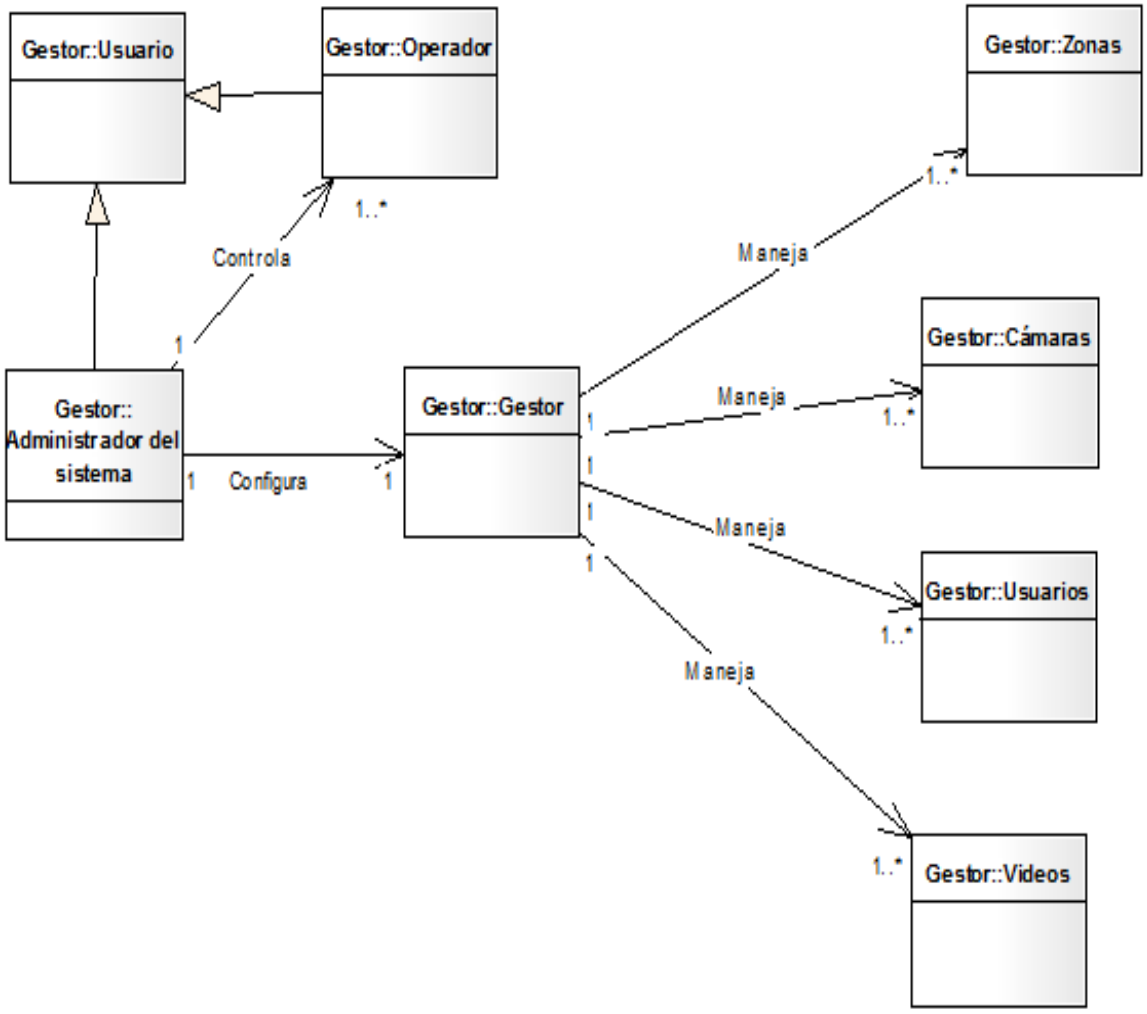

Figura1: Diagrama del modelo de Dominio.

2.2-REQUERIMIENTOS FUNCIONALES DEL SISTEMA.

A continuación se enumeran los requerimientos funcionales identificados y obtenidos a partir de las necesidades del sistema, ellos son los que describen

con detalle la función de éste, sus entradas y salidas, excepciones, y además definen los alcances del trabajo en cuanto a las acciones que debe realizar. El sistema propuesto consta de once requerimientos de ellos ocho son gestionar compuestos por las 3 operaciones fundamentales: Adicionar, modificar y eliminar los otros tres son: Registrar logs, Enviar información y Servir datos.

2.3.1-Requerimientos funcionales del Gestor:

RF1 Gestionar Cámara
RF1.1 Adicionar cámara
RF1.2 Modificar cámara
RF1.3 Eliminar Cámara
RF1.4 Obtener cámaras.

RF2 Gestionar Zonas Físicas.
RF2.1 Adicionar Zona.
RF2.2 Editar Zona.
RF2.3 Eliminar Zona.

RF3 Gestionar la conexión a la base de datos. (logeo).

RF4 Gestionar regla de grabación.
4.1 Adicionar regla de grabación.
4.2 Actualizar regla de grabación.
4.3 Eliminar regla de grabación.

RF5 Registrar logs de Actividad.

RF6 Gestionar usuarios.
RF6.1 Insertar usuario.
RF6.2 Actualizar usuario.
RF6.3 Eliminar usuario.

RF7 Gestionar Roles.

RF7.1 Obtener roles.

RF7.2 Insertar rol.

RF7.3 Actualizar rol.

RF7.4 Eliminar rol.

RF8 Gestionar videos almacenados.

8.1 Insertar datos de video.

8.2 Actualizar datos de video.

8.3 Eliminar datos de video.

RF9 Gestionar Mapas de grabación.

9.1 Insertar mapa de grabación.

9.2 Actualizar mapa de grabación.

9.3 Eliminar mapa de grabación.

R.F.10 Enviar Información.

El Sistema debe enviar toda la información necesaria en dependencia de como la soliciten los clientes de los restantes módulos (En este caso el cliente que se implementa para realizar las pruebas parciales).

R.F.11 Servir Datos.

El cliente gestor debe realizar las consultas necesarias al gestor para satisfacer La información solicitada por los demás módulos.

2.3 -REQUERIMIENTOS NO FUNCIONALES DEL SISTEMA:

En este trabajo de diploma los RNF se encuentran en estrecha vinculación a los requisitos funcionales, ya que se hace necesario especificar cómo el sistema debe realizar dichos RF y que debe cumplir para ello, una vez que dichos RF han sido definidos y implementados.

2.3.1-SOFTWARE:

RF1 Windows XP o superior.

RF2 Qt framework version 4.7.0.

2.3.2-HARDWARE:

RF31.5 Gb de RAM.

RF4 Procesador Pentium 4 a 3.0GHz o superior.

2.3.3-SOPORTE:

RNF5 Se necesita el QT (QT creator) 4.7.0 o superior instalado en todas las estaciones de Trabajo.

RNF6 Se necesita Windows XP SP2 o superior instalado en todas las estaciones de Trabajo.

2.3.4-SEGURIDAD:

RNF8 Se garantizará la seguridad del Sistema mediante el uso de varios usuarios con varios niveles de privilegios distintos protegidos por contraseña.

RNF9CONFIDENCIALIDAD: La información manejada por el sistema está protegida de acceso no autorizado y divulgación.

RNF10INTEGRIDAD: la información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes.

RNF11DISPONIBILIDAD: Significa que los usuarios autorizados se les garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

2.3.5-CONFIABILIDAD:

RNF12 El sistema debe ser confiable desde todo punto de vista debido a la sensibilidad de la información que maneja.

2.3.6-APARIENCIA E INTERFAZ EXTERNA:

RNF13 La interfaz debe ser sencilla y cómoda de manera que se maximice el área expositiva sin muchos colores o formas y que permita un fácil manejo para el administrador y el operador.

2.3.7-LEGALES:

RNF14 El sistema deberá ser regido por todas las normas, leyes, regulaciones y estipulaciones que se relacionen con él y sus componentes.

2.4-DEFINICIÓN DE CASOS DE USO:

2.4.1-DEFINICIÓN DE ACTORES:

En este caso se cuenta con 5 actores fundamentales (Administrador, Operador, Usuario, Gestor, Cliente gestor).Lo que define y modela a los actores del sistema de este trabajo de diploma es el rol que estos juegan, dicho rol es el que muestra cómo se comunican entre sí y con los CU.

Actor	**Descripción**
Administrador	Rol responsable de velar por el funcionamiento apropiado del sistema, configurarlo y realizar todas las demás funcionalidades que permite el sistema.
Operador	Rol que se beneficia de diversas funcionalidades del sistema pero no tiene acceso a las configuraciones del mismo.
Usuario	Rol que representa una generalización especialización de los actores Administrador y Operador.
Gestor	Módulo encargado de la gestión de datos del sistema así como de su manipulación a nivel de Base de datos.
Cliente Gestor	Es el encargado de obtener los datos desde el servidor levantado por el gestor.

2.4.2-Casos de usos:

Los CU que se detectaron fueron 11 los mismos son descritos y enumerados a continuación.

CU-1	Gestionar Cámaras
Actor	Administrador
Descripción	El Administrador adiciona, modifica, elimina u obtiene una

	cámara.
Referencia	RF1.1, RF1.2, RF1.3, RF1.4

CU-2	Gestionar Zonas Físicas
Actor	Administrador
Descripción	El Administrador adiciona, edita, elimina u obtiene zonas que representan una ubicación física en la instalación donde se encuentra desplegado el Sistema.
Referencia	RF2.1, RF2.2, RF2.3, RF 2.4

CU-3	Gestionar conexión con la Base de Datos(logeo)
Actor	Administrador
Descripción	Permite configurar los parámetros de la conexión con la base de datos así como logearse a los usuarios del sistema.
Referencia	RF3,RF3.1

CU-4	Gestionar regla de grabación.
Actor	Administrador
Descripción	El Administrador adiciona, modifica, elimina u obtiene una regla de grabación.
Referencia	RF4. RF4.1, RF4.2, RF4.3.

CU-5	Registrar Logs de Activated.
Actor	Gestor.
Descripción	Se registran logs de la actividad de cada instancia de los módulos con respecto a la información almacenada en el Sistema.

Referencia	RF5.

CU-6	Gestionar Usuarios
Actor	Administrador
Descripción	El Administrador gestiona (lo inserta, actualiza y elimina) un usuario.
Referencia	RNF6,RF6.1,RF6.2,RF6.3

CU-7	Gestionar Roles.
Actor	Administrador
Descripción	El Administrador adiciona, modifica, elimina u obtiene un Rol.
Referencia	RNF7,RF7.1, RF7.2, RF7.3.

CU-8	Gestionar videos almacenados.
Actor	Administrador
Descripción	El Administrador adiciona, modifica, elimina u obtiene los datos de un video.
Referencia	RF8.1, RF8.2, RF8.3.

CU-9	Gestionar videos mapas de grabación.
Actor	Administrador
Descripción	El Administrador adiciona, modifica, elimina u obtiene los mapas de grabación.
Referencia	RF9.1, RF9.2, RF9.3

CU-10	Enviar Información.
Actor	Gestor

Descripción	Permite obtener los datos de la bd mediante el cliente de las pruebas parciales.
Referencia	RNF10

CU-11	Servir Datos (Consulta).
Actor	Gestor.
Descripción	Permite realizar las consultas necesarias al gestor para Satisfacer los datos solicitados por los demás módulos.
Referencia	RNF11

2.5-DIAGRAMA DE CU:

En este trabajo de diploma el diagrama de casos de usos queda compuesto por 11 CU identificados previamente y unidos a un actor por la función que estos realizan, además se utilizan para ilustrar los requerimientos del sistema al mostrar cómo este reacciona a eventos que se producen en su ámbito y como dichos eventos se relacionan con un actor especifico.

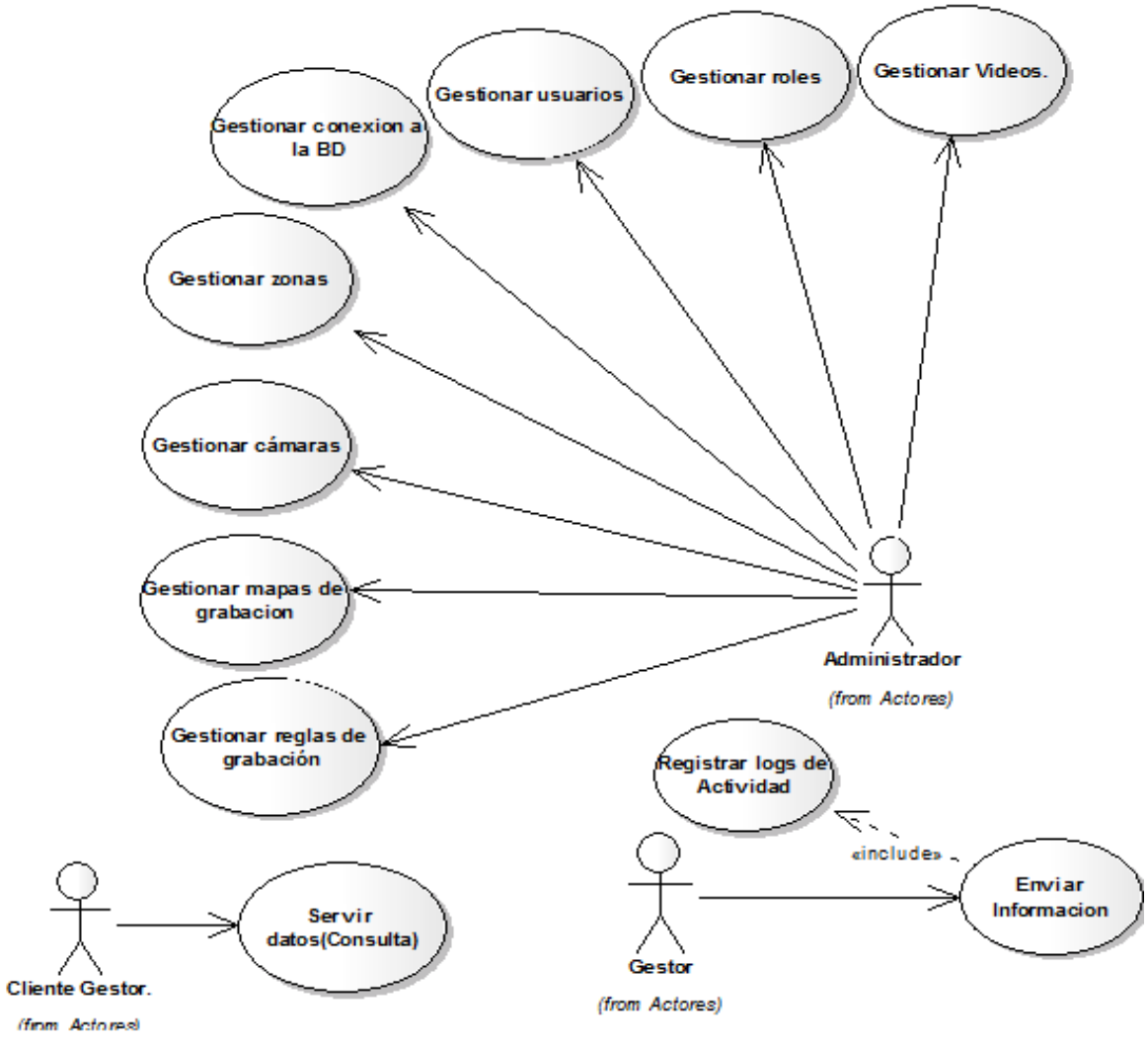

Figura 2: Diagrama de casos de uso.

2.6-CASOS DE USOS POR CICLO:

Aquí se enumeran los Casos de uso haciendo énfasis en los ciclos en los cuales los mismos van a ser desarrollados, de esta forma de trabajar se obtiene buenos resultados ya los CU no se realizan todos a la vez sino que se sigue un orden de prioridad para su implementación priorizándose los más críticos, en este caso 9 para el primer ciclo de desarrollo y solo dos para el segundo.

2.6.1-PRIMER CICLO DE DESARROLLO:

Código	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-1	Gestionar Cámaras	Gestor	Es una de las funcionalidades básicas

			del Sistema.
CU-2	Gestionar Zonas	Gestor	Es una de las funcionalidades básicas del Sistema.
CU-4	Gestionar regla de grabación	Gestor	Es una de las funcionalidades básicas del Sistema.
CU-6	Gestionar usuarios.	Gestor	Es una de las funcionalidades básicas del Sistema.
CU-7	Gestionar roles.	Gestor	Es una de las funcionalidades básicas del Sistema.
CU-8	Gestionar videos almacenados.	Gestor	Es una de las funcionalidades básicas del Sistema.
CU-9	Gestionar mapas de grabación.	Gestor	Es una de las funcionalidades básicas del Sistema.
CU-11	Servir datos.	Gestor	Es una de las funcionalidades básicas del sistema.
CU-10	Enviar información.	Gestor	Es una de las funcionalidades básicas del sistema.

2.6.1-SEGUNDO CICLO DE DESARROLLO:

Código	**Nombre de caso de uso**	**Paquete**	**Justificación de la selección.**
CU-5	Registrar Logs de Actividad.	Gestor	No es una funcionalidad esencial durante el desarrollo.
CU-3	Gestionar la conexión Base de Datos(logeo)	Gestor	No es una funcionalidad esencial durante el desarrollo.

2.7-CASOS DE USOS EXPANDIDOS:

Luego de ser identificados los CU de este sistema y ser mostrados los mismos con su diagrama de CU, se procede a realizar su descripción basándose en las principales acciones que puede ejecutar los actores del sistema (Administrador, Operador, Cliente Gestor y Gestor) y la consecuente respuesta del sistema, para ello se definen los escenarios específicos por los cuales los CU pueden transitar, así como su prototipo de interfaz. Debido a su gran tamaño solo se muestra la descripción completa de solo un caso de uso (Gestionar cámaras) resumiéndose los demás.

1.-Gestionar cámaras.

<table>
<tr><td>Caso de Uso:</td><td colspan="2">Gestionar cámaras</td></tr>
<tr><td>Actores:</td><td colspan="2">Visor</td></tr>
<tr><td>Resumen:</td><td colspan="2">Permite adicionar, actualizar y eliminar las cámaras del sistema y termina el CU cuando se almacenan los datos de la base de datos.</td></tr>
<tr><td>Referencias</td><td colspan="2">RF1.1, RF1.2., RF1.3.,RF1.4</td></tr>
<tr><td>Prioridad</td><td colspan="2">Alta</td></tr>
<tr><td colspan="3">Flujo Normal de Eventos</td></tr>
<tr><td colspan="3">Sección “Adicionar cámara.”</td></tr>
<tr><td colspan="2">Acción del Actor</td><td>Respuesta del Sistema</td></tr>
<tr><td colspan="2">1. El administrador selecciona la opción tbl_camaraip y marca la pestaña Insertar en la nueva interfaz.

3. El administrador llena los campos con los datos que correspondan y presiona el botón aceptar.</td><td>2. El sistema muestra una interfaz de con los campos vacíos necesarios para insertar una cámara.

4. El caso de uso termina cuando se almacenan los datos de la nueva cámara en la Base de Datos.</td></tr>
</table>

Sección "Actualizar cámara."	
Acción del Actor	**Respuesta del Sistema**
1.-El administrador da click en la pestaña modificar. **3. -El administrador selecciona el id de la cámara, así como el campo de la misma que va a modificar.** **5.-El administrador introduce los datos y presiona el botón aceptar.**	**2.- El sistema muestra los id de las cámaras existentes en la bd.** **4.-El sistema muestra cómo introducir los datos a ser modificados.** **6. El caso de uso termina cuando se actualizan los datos de la cámara en la Base de Datos.**
Sección "Eliminar cámara."	
Acción del Actor	**Respuesta del Sistema**
1. El administrador da click en la pestaña eliminar. **3. El administrador selecciona la cámara que desea eliminar y presiona el botón aceptar.**	**2. El sistema muestra cómo seleccionar el id de la cámara a ser eliminada mediante un combo Box.** **4. El caso de uso termina cuando se eliminan los datos de la cámara en la Base de Datos.**

2.-Gestionar zonas.

Caso de Uso:	**Gestionar Zonas físicas**
Actores:	**Administrador**
Resumen:	**Permite adicionar, actualizar y eliminar las zonas del sistema y concluye cuando las mismas son almacenadas en la base de datos.**
Referencias	**RF2.1, RF2.2., RF2.3.**

3.-Gestionar la conexión a la base de datos.

Caso de Uso:	**Gestionar conexión con la base de datos (Logeo).**
Actores:	**Administrador.**
Resumen:	**Permite configurar los parámetros de la conexión con la base de datos así como logearse a los usuarios del sistema (Databasename, Password, user).**
Referencias	**RF3.0**

4.-Gestionar regla de grabación.

Caso de Uso:	**Gestionar regla de grabación.**
Actores:	**Administrador.**
Resumen:	**Permite adicionar, actualizar y eliminar las reglas de grabación y concluye cuando las mismas son almacenadas en la base de datos.**
Referencias	**RF4.1, RF4.2., RF4.3.**

5.-Registrar logs de Actividad.

Caso de Uso:	**Registrar logs.**
Actores:	**Gestor.**
Resumen:	**Permite almacenar todos los logs de las operaciones que se realizan en el sistema dentro de un archivo *.txt y cuya dirección es especificada al logearse algún usuario del gestor.**
Referencias	**RF5**

6.-Gestionar usuarios.

Caso de Uso:	**Gestionar usuarios.**
Actores:	**Administrador.**
Resumen:	**Permite adicionar, actualizar y eliminar los usuarios y termina cuando se guardan los datos de los usuarios en la base de datos.**
Referencias	**RF6.1, RF6.2., RF6.3.**

7.-Gestionar roles.

Caso de Uso:	**Gestionar roles.**
Actores:	**Administrador.**
Resumen:	**Permite adicionar, actualizar y eliminar los roles del sistema y concluye cuando los mismos son almacenados en la base de datos.**
Referencias	**RF7.1, RF7.2., RF7.3.**

8.-Gestionar videos.

Caso de Uso:	**Gestionar videos almacenados.**
Actores:	**Administrador.**
Resumen:	**Permite adicionar, actualizar y eliminar los datos de los videos del sistema y concluye cuando los mismos son guardados en la base de datos.**
Referencias	**RF8.1, RF8.2., RF8.3.**

9.-Gestionar mapas de grabación.

Caso de Uso:	**Gestionar mapas de grabación.**
Actores:	**Administrador.**
Resumen:	**Permite adicionar, actualizar y eliminar los datos de los mapas de grabación del sistema y se termina cuando los mapas son guardados en la base de datos.**
Referencias	**RF9.1, RF9.2., RF9.3.**

10.-Enviar Información.

Caso de Uso:	**Enviar Información**
Actores:	**Gestor**
Resumen:	**El actor envía toda la información según la soliciten los clientes de los demás módulos (En este caso el cliente de las pruebas parciales).**
Referencias	**RF10**

11.-Servir Datos.

Caso de Uso:	**Servir datos (Consulta).**
Actores:	**Cliente gestor**
Resumen:	**El cliente de las pruebas parciales realiza la solicitud de datos necesarios para probar su conexión con el servidor gestor que se encuentra conectado a la bd.**
Referencias	**RF11**

2.8–CONCLUSIONES:

En este capítulo se ha obtenido como resultado el modelo del dominio del sistema que muestra los principales objetos y roles del mismo, el modelo de dominio es usado por el implementador como una forma de obtener una visión más clara del sistema, también en el marco de este capítulo se ha definido los requerimientos funcionales y no funcionales que han sido identificados dentro del ámbito del sistema dichos requerimientos son esenciales en especial los funcionales ya trazan el camino a seguir de acuerdo a lo que se espera que el software o aplicación a desarrollar deba cumplir en dependencia de las necesidades del usuario.

Por otro lado a lo largo de este capítulo se obtuvo como resultado los actores del negocio que nos ocupa, los casos de usos son también parte del resultado de esta parte del documento ya que ellos se definen y se obtienen fundamentalmente a partir de los requerimientos funcionales y constituyen las operaciones que debe realizar el sistema para satisfacer las necesidades de un actor.

Y además por último y no menos relevante se estableció el diagrama de CU del sistema basado en los mismos Casos de Usos obtenidos con anterioridad estableciendo las relaciones que vinculan a los CU con los actores más relevantes del sistema, posteriormente se describen los CU por ciclo de desarrollo y los CU de manera expandida donde se arroja la estructura interna de los mismos de manera más profunda y concreta y que se espera que ellos realicen concretamente.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.

INTRODUCCIÓN

En este capítulo se realiza el diseño de la propuesta de la solución exponiendo las principales características y mostrando todas las propiedades que componen al sistema, para esto se lleva a cabo cada uno de los diagramas de clases del diseño en correspondencia con cada caso de uso que ha sido identificado previamente, además se define la arquitectura primaria del trabajo haciendo énfasis en su estructura global así como describiéndola detalladamente y por último se realiza el modelo de diseño del trabajo el cual es de vital importancia para el mismo.

3.1-AQUITECTURA Y PROPUESTA DE SOLUCIÓN.

La definición de la arquitectura de software es según Pressman:*"la estructura jerárquica de los módulos del programa, la manera de interactuar de estos componentes, y la estructura de los datos usados por estos módulos"* (13).

El Sistema se encuentra diseñado para seguir una arquitectura primaria basada en la arquitectura de pizarra en su variante de tablero de control. Esta consiste en desarrollar el sistema sobre la base de componentes que son llamados agentes autónomos, ellos realizan todas sus funcionalidades de forma independiente. Pero a su vez dependen de la entrada de información externa es decir de información que necesitan y solicitan para su correcto funcionamiento, que es provista a su vez por otros agentes, y que conducen a un resultado en específico que puede a su vez ser entrada de otros agentes que lo necesiten(1).

Mediante esta forma de trabajar se obtiene buenos resultados en la centralización del Sistema. Cada agente se rige por patrones y estándares que permiten que ocurran cambios en el ambiente externo al agente sin que este sufra cambios en su funcionamiento interno. El funcionamiento de todos los agentes autónomos se coordina por un elemento central, este se comúnmente se denomina Repositorio Activo de información, el mismo entrega y recibe

información de los agentes y coordina su funcionamiento entre sí. El Gestor en este caso sería el Repositorio Activo, y los demás módulos del sistema distintos agentes autónomos que funcionarían en coordinación con el Sistema. 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.

Aunque el Sistema mantiene una Arquitectura de Pizarra a manera global, en cada uno de los módulos se usa el estilo Modelo-Vista-Controlador (MVC) y el de 3 Capas.

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Modelo: Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

El Modelo no se inmuta ante los cambios en la interfaz de usuario, el mismo se encarga de toda la gestión interna del negocio, la Vista se encarga de presentar los datos y de interactuar externamente con el usuario, mientras que el controlador maneja todos los cambios de estado de la Vista. Mediante el uso de este modelo los módulos del sistema son capaces de usar diferentes interfaces de usuario sin que se modifique la lógica de negocio. Se distribuye todo el diseño del módulo en 3 capas bien definidas, cada una de ellas agrupan clases con objetivos muy específicos dirigidos a una meta en común. En el caso del Sistema se usa para delimitar las capas de Presentación, Negocio y Acceso a Datos.

PROPUESTA PARA EL SISTEMA:

La solución que se plantea es la de desarrollar un sistema de gestión, que proveerá las funcionalidades necesarias para el manejo de los videos e imágenes, así como de los datos de las cámaras. El Gestor es el que provee la información a cada instancia de los módulos del sistema. El Gestor está compuesto por una capa de negocio y una de acceso a datos.

Este debe de estar siempre activo en el sistema, y desde que se inicia mantiene una lista de todas las cámaras en el sistema, de su configuración y todo tipo de información necesaria para el funcionamiento de cada una de ellas, información que obtiene de la Base de Datos. El Gestor para cada cliente registra información de su actividad, y del usuario que llevo a cabo esa actividad realizada, y asigna un identificador temporal a ese cliente, que lo distinguirá de las demás estaciones del sistema y le permitirá realizar operaciones futuras con el Gestor.

Del Gestor se descargará toda la información relacionada a las cámaras y otros elementos de configuración necesarios para su funcionamiento. El operador podrá entonces manipular los videos e imágenes, así como las cámaras y hacer cambios sobre estas, si es un administrador. Cualquier modificación a los datos existentes se le solicita al Gestor que la procesa y avisa a las demás instancias del sistema que se encuentren activas, para que reflejen dicho cambio.

Es de frecuente interés para los operadores que se encuentran en las estaciones de visualización y gestión la necesidad de juntar o agrupar un diverso número de cámaras porque están relacionadas de alguna manera (entre todas visualizan un área determinada, o un edificio en particular). En el sistema se propone el concepto de zonas donde a cada zona se le asociaría los datos pertenecientes a un grupo del sistema cuyos datos se obtienen de la base de datos y que a su vez posen un conjunto de cámaras asociadas a él.

3.2-MODELO DE ANÁLISIS:

El análisis consiste en obtener una visión más clara del sistema, el mismo se preocupa de ver que es lo que hace como tal el mismo. El modelo de análisis constituye la entrada principal para el comienzo del diseño del sistema. La tarea de la realización del modelo de análisis permite al desarrollador o desarrolladores especificar el flujo de datos y control del sistema, además de identificar los elementos básicos del sistema.

El objetivo del análisis por lo tanto, se centra en comprender perfectamente los requisitos del software y no precisar cómo se implementa el producto. En el modelo del análisis se refinan los requisitos, no se toman en cuenta el lenguaje de programación a usar en la construcción, la plataforma en la que se ejecuta la aplicación, los componentes reutilizables de otras aplicaciones, entre otras características que afectan al sistema.

Diagrama de clases de análisis del sistema.

A continuación sólo se muestran los diagramas de algunos de los CU más significativos para este sistema.

Los diagramas de clases del análisis para este proyecto se realizan para cada caso de uso del sistema y muestra las principales clases participantes en dicho caso de uso y sus relaciones. En estos diagramas se identifican las principales clases que compondrán este sistema gestor divididas en: Interfaz, Controladora y Entidad.

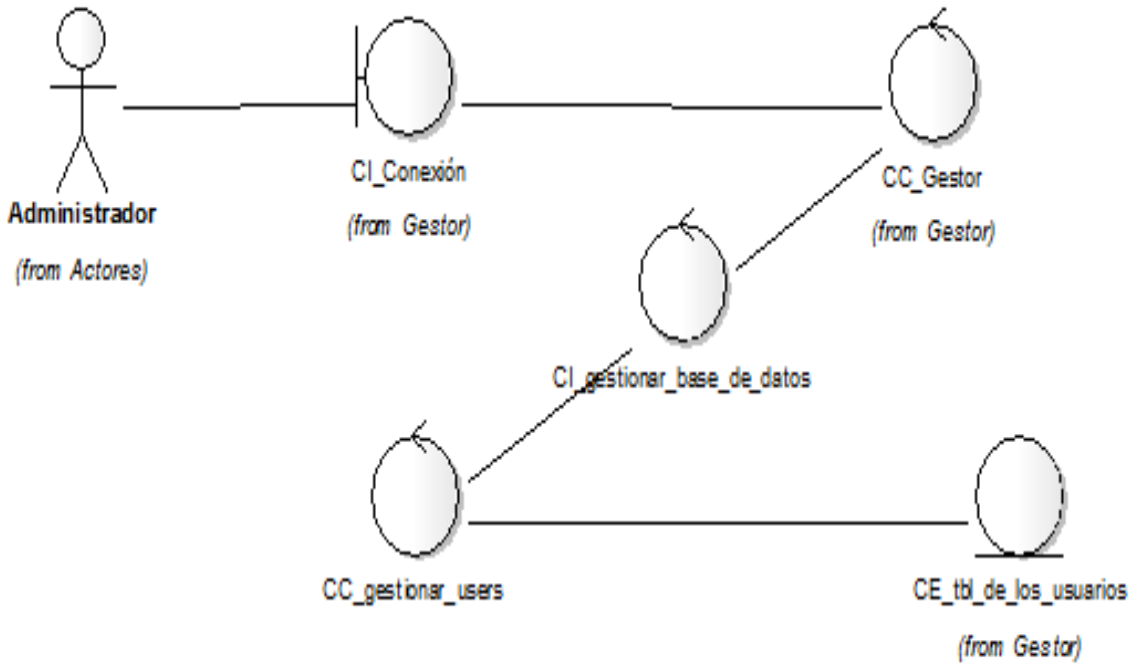

Figura 3: CU Gestionar usuario.

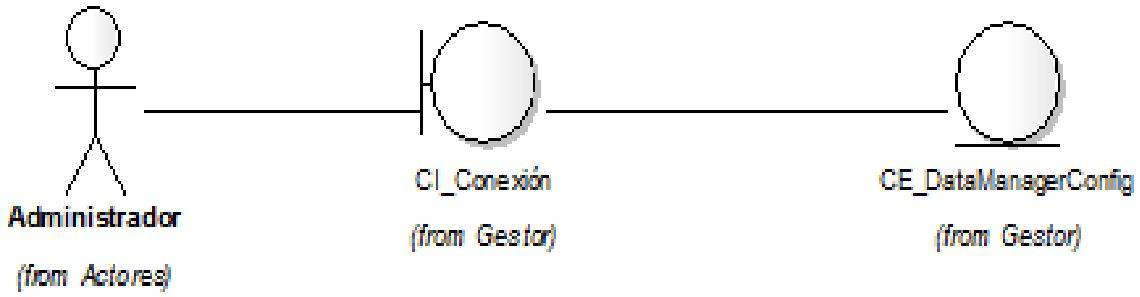

Figura 4: CU Gestionar la conexión con la BD.

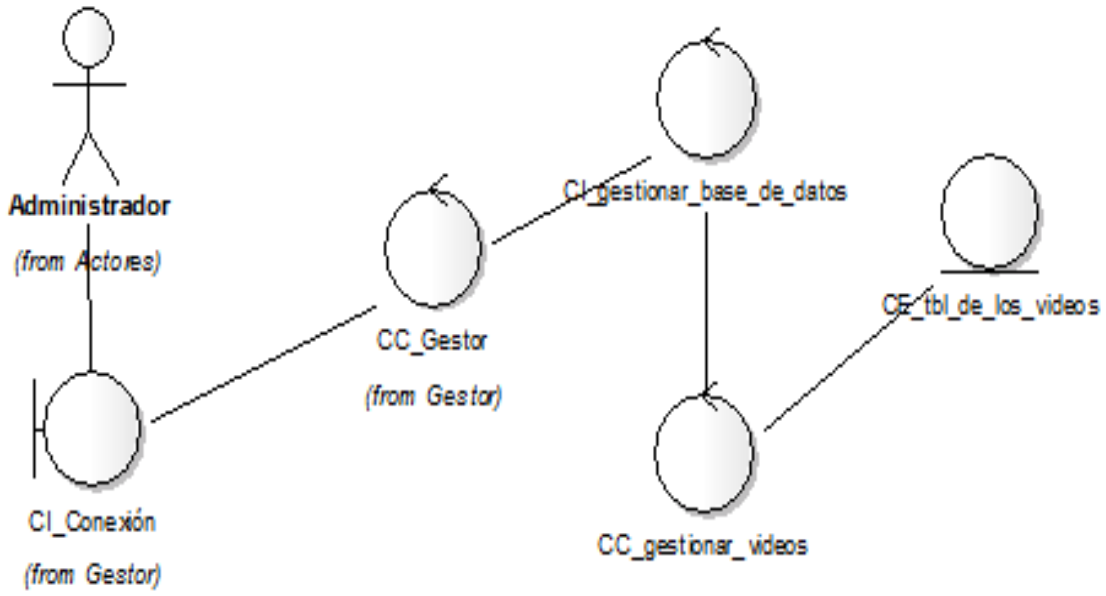

Figura 5: CU Gestionar los videos.

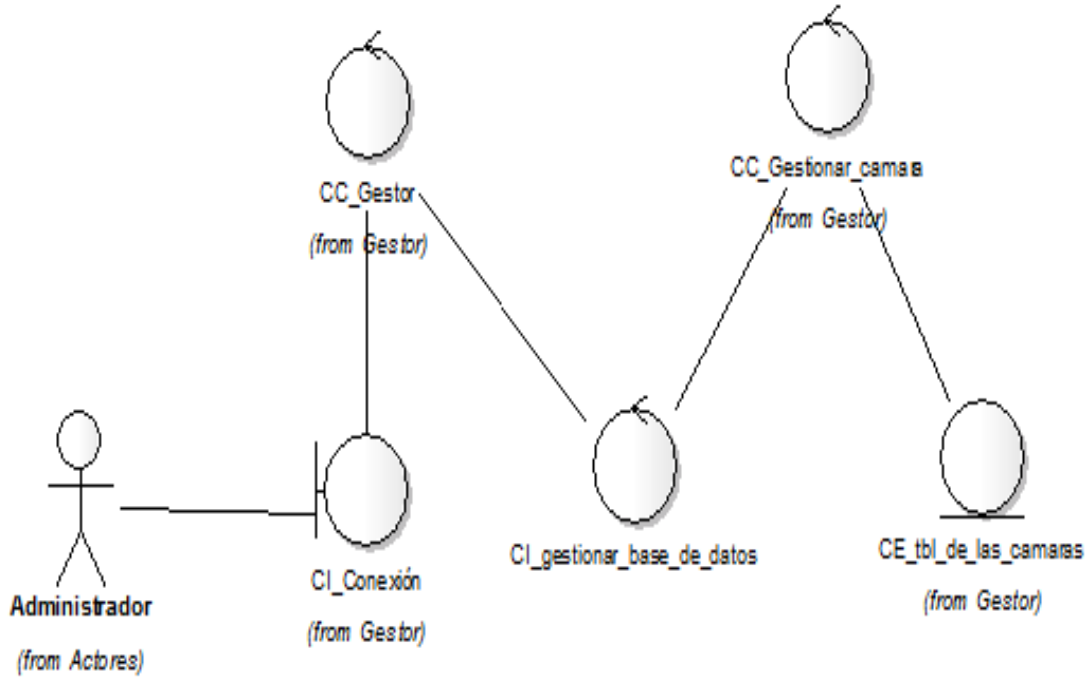

Figura 6: CU Gestionar Cámaras.

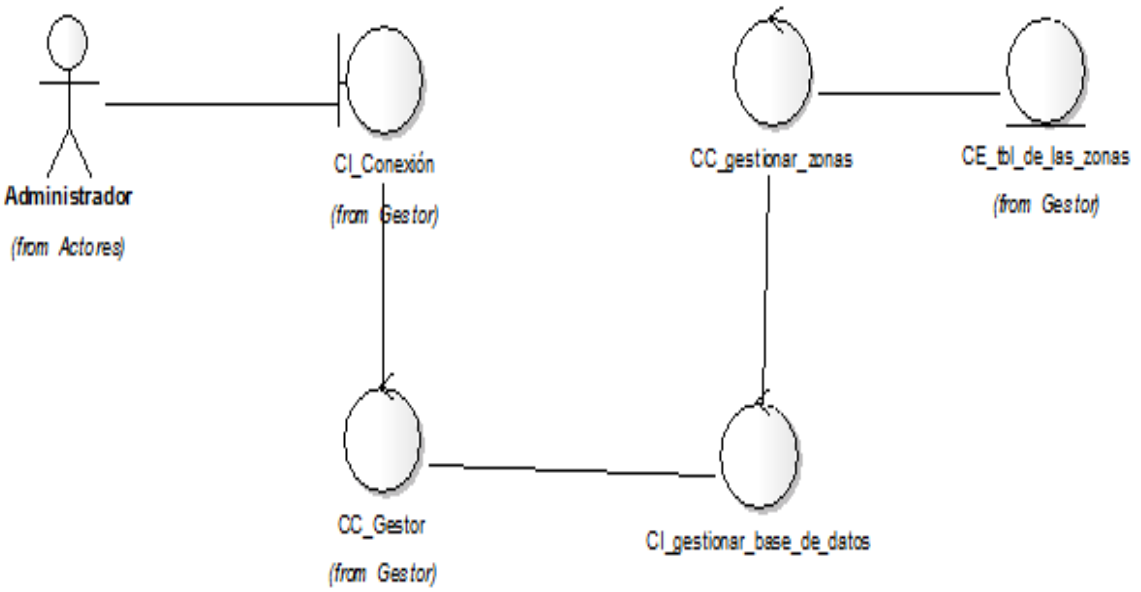

Figura 7: CU Gestionar Zonas.

3.3-MODELO DEL DISEÑO.

En esta etapa de desarrollo es cuando los requerimientos funcionales se traducen en representaciones de clases de software, el diseño es el primer paso para la implementación de cualquier software, *"es producir un modelo o representación de una entidad que se va a construir posteriormente"* según Pressman (13).

A continuación se presentan los diagramas de clases de algunos de los casos de uso de este sistema en específico así como las descripciones de las principales clases involucradas en ellos. El Modelo de Diseño de este trabajo se compone por el diagrama de clases de diseño que se encarga de describir de manera gráfica las especificaciones de las clases de software y de las interfaces de la aplicación (el módulo gestor), . El Modelo de Diseño de este sistema por lo tanto es el que define y describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el mismo.

Estos diagramas están compuestos por: Clases, atributos, las asociaciones entre las clases.

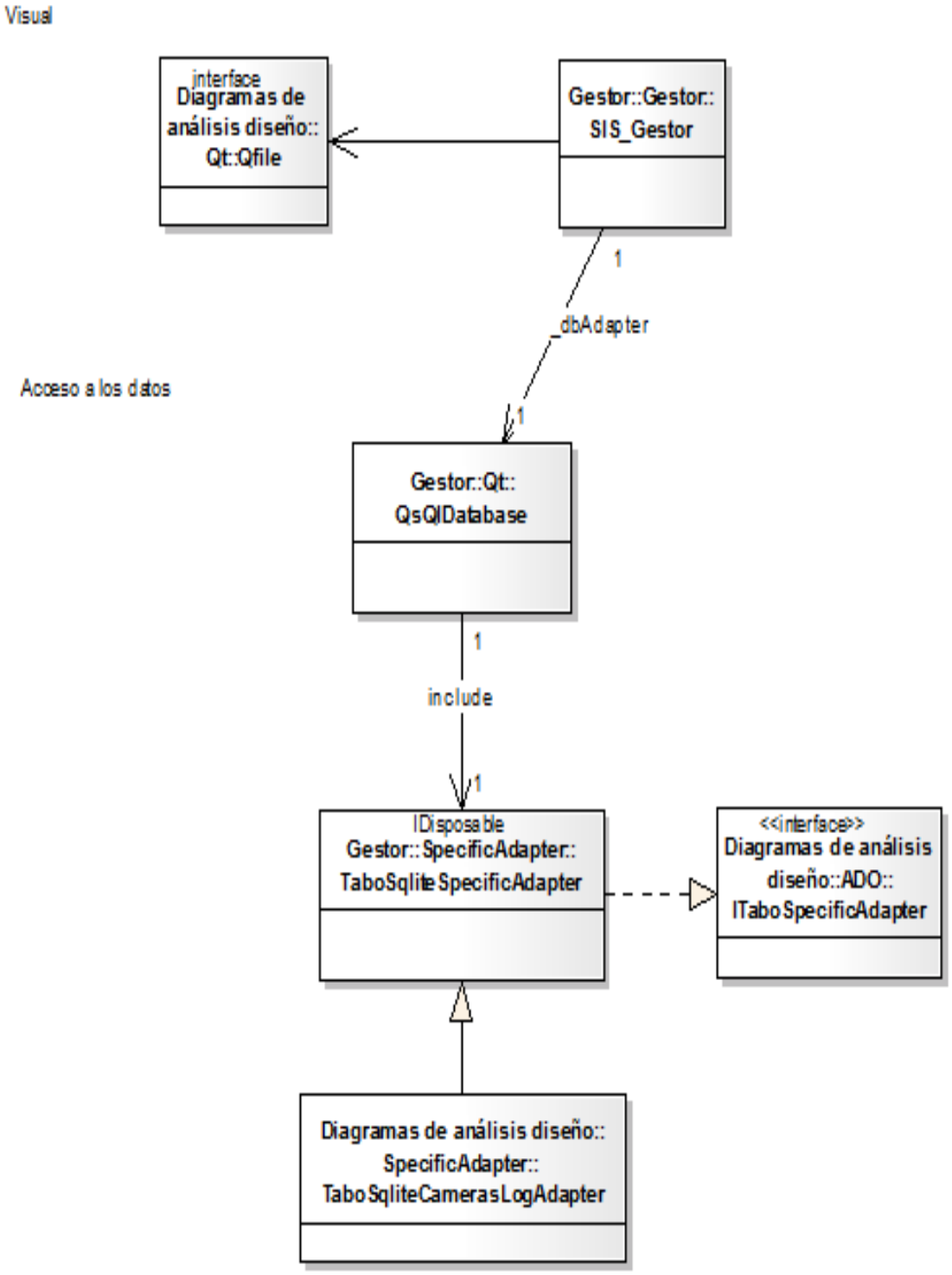

Figura 8: Diagrama de clases CU Registrar Logs de Actividad.

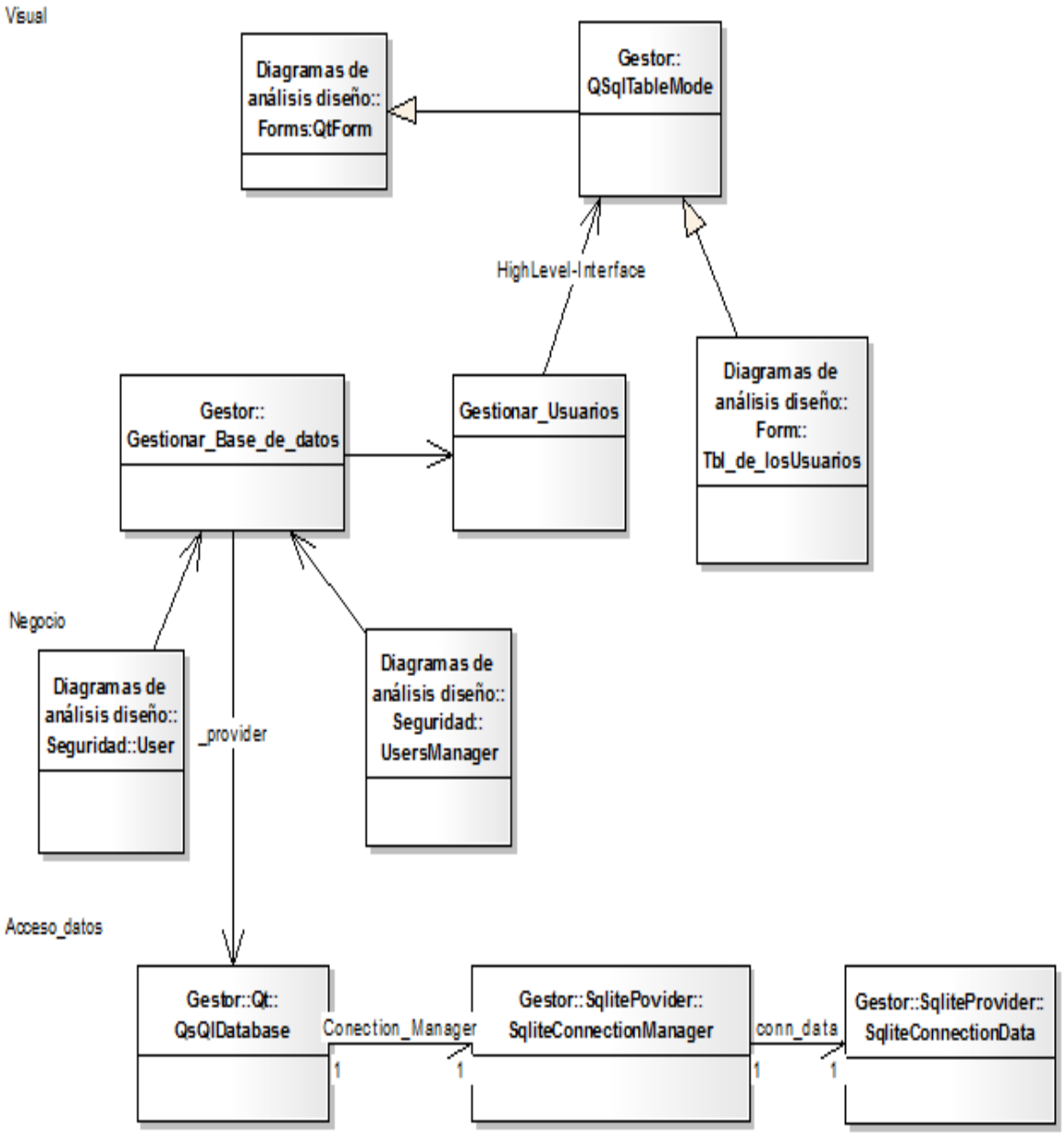

Figura 9: Diagrama de clases CU Gestionar Usuarios.

3.4-DESCRIPCIÓN DE LAS CLASES:

Aquí se hace énfasis en la especificación de la estructura interna de algunas de

las clases mostrando su conjunto de atributos, operaciones y relaciones para una mayor compresión de las mismas.

Nombre: Gestor	
Tipo de clase : Controladora	
Atributo	**Tipo**
_logActividad	QString
_Name	QString
_Password	QString
_nameBD	QString
_nameuser	int
_nombreRol	QString
_ a	Server
Para cada responsabilidad:	
Nombre:	crearServer()
Descripción:	Crea el servidor que envía los datos al cliente de la base de datos del gestor.
Nombre:	Gestionar()
Descripción:	Crea la interfaz para la gestión de los datos de la base de datos.
Nombre:	Logout()
Descripción:	Permite deslogearse del gestor para cambiar de usuario.
Nombre:	ReturnConexion(DataManagerConfig* datos)
Descripción:	Permite obtener el nombre de la base de datos para su manejo y gestión a partir de la clase datamanegerconfig.

Nombre: Group	
Tipo de clase : Entidad	
Atributo	**Tipo**

description	string
Id	int
name	string
parent	Group
Time	DateTime

Para cada responsabilidad:	
Nombre:	Clone()
Descripción:	Devuelve un clon del grupo en una nueva instancia.
Nombre:	CopyFrom(Group)
Descripción:	Copia los datos desde un grupo ya creado.
Nombre:	Description()
Descripción:	Devuelve una descripción del grupo.
Nombre:	FullName()
Descripción:	Devuelve el nombre completo del grupo.
Nombre:	Group(string)
Descripción:	Crea una nueva instancia.
Nombre:	ID
Descripción:	Devuelve el ID del grupo
Nombre:	Name
Descripción:	Devuelve el nombre del grupo
Nombre:	Parent
Descripción:	Devuelve el padre del grupo.

Nombre: User	
Tipo de clase : Entidad	
Atributo	**Tipo**

_name	string
_password	string
IsAdmin	bool
Para cada responsabilidad:	
Nombre:	User(string Nombre, string Password ,boolIsAdmin)
Descripción:	Constructor que crea el Usuario con los datos requeridos.

Nombre: Conexion	
Tipo de clase : Interfaz	
Atributo	**Tipo**
g	Gestor
db	QSqlDatabase
altura	int
anchura	int
size	int
contador	int
Para cada responsabilidad:	
Nombre:	Conexion ()
Descripción:	Constructor que inicializa la conexión con los datos requeridos.
Nombre:	on_pushButton_clicked()
Descripción:	Permite chequear que la bd, el usuario y el password sean correctos y de esta manera inicializar el gestor, en caso contrario muestra un cuadro de dialogo con el error correspondiente.
Nombre:	on_pushButton_2_clicked()
Descripción:	Cancela la ventana cerrándola.
Nombre:	conectar(QString name)
Descripción:	Permite comprobar que el name del usuario de la base de datos coincide con el que se entró en el lineEdit_2.

Nombre:	conectarp(QString password)
Descripción:	Permite comprobar que el password del usuario de la base de datos coincide con el que se entró en el lineEdit_3.
Nombre:	chequearUSerVSPassword (QString user, QString password)
Descripción:	Permite chequear en caso que el user y el password efectivamente existan en la BD, que los mismos son de un mismo usuario y no de usuarios distintos.

3.5–CONCLUSIONES:

Este capítulo ha arrojado como resultado la arquitectura de software definida para la aplicación, esto es de vital importancia ya que ella es esencial para el desarrollo del sistema y a su vez es la encargada de definir el funcionamiento interno del mismo, así mismo en esta sección del documento se ha obtenido la propuesta de solución, la misma evidencia todo lo que se espera del sistema en sí, y como este funcionará en coordinación con los otros módulos.

Por otro lado en este capítulo se ha obtenido como parte del mismo el modelo de clases de análisis, el cual es muy importante porque es un eslabón fundamental para obtener una visión más clara del negocio, así mismo este modelo se encarga de dejar en claro los componentes básicos del sistema, posteriormente a los diagramas de clases de análisis en esta parte se obtuvo como resultado a partir de los requerimientos funcionales el modelo del diseño que es donde se muestran los Requisitos Funcionales como clases de software, los mismos quedan descritos en las clases del diseño que son indispensables a la hora de mostrar las interacciones entre las distintas partes del sistema, así como el momento en que estas ocurren y las asociaciones que las entrelazan. Y por último este capítulo enseña como resultado las descripciones de las clases con algunos de sus atributos, relaciones, y métodos.

A manera de cierre se ha considerado que este capítulo es muy importante para este ciclo de desarrollo porque el mismo detalla elementos esenciales descritos ya anteriormente que constituyen entradas fundamentales para ser utilizadas en el correcto desarrollo del inicio de la implementación.

CAPITULO 4: IMPLEMENTACIÓN Y PRUEBAS.

INTRODUCCIÓN

En este capítulo de la implementación se comienza a desarrollar la misma a partir del modelo y diseño del sistema, en términos de componentes es decir ficheros de códigos fuentes y binarios, ejecutables, etc. Aquí se detalla el modelo de datos donde se ve la estructura que van a tener dichos datos en la aplicación, posteriormente se muestran los diagramas de despliegue y componentes que conforman lo que se conoce como modelo de implementación, estos diagramas muestran los componentes a desarrollar y construir, la estructura de los mismos y también las relaciones de dependencia entre los nodos sobre los que estará basada la aplicación.

A continuación se muestra el modelo de datos e implementación de este software como resultado del diseño realizado con anterioridad.

4.1-MODELO DE DATOS:

El modelo de datos en este trabajo define y describe la estructura de los mismos en la base de datos. El mismo se encarga de mostrar como ellos se vinculan entre sí. De manera general se puede decir que los modelos de datos se basan en conceptos claves como entidades, atributos y relaciones, en esta bd se tiene que cuenta con 12 tablas compuestas por los campos que permiten las operaciones de manipulación que se pueden realizar con los mismos (agregado, borrado, modificación y recuperación de los datos).

A manera de resumen se puede decir que en este sistema la base de datos es la que proporciona cierto nivel de abstracción de datos, es decir el modelo de datos es algo que la mayoría de usuarios no necesita conocer, a continuación se muestran las tablas que componen el mismo.

Nombre: tbl_camaraip		
Descripción: Representa una cámara en el Sistema.		
Atributo	**Tipo**	**Descripción**
id	integer	ID de la cámara.
tipo_camara	varchar(20)	Tipo de la cámara.
fabricación	varchar(50)	Fabricante
nombre	varchar(80)	Nombre de la cámara.
ip	varchar(255)	IP de la cámara.
provider	varchar(100)	Provider de la cámara.
config_provider	varchar(255)	Config_provider de la cámara
descripcion	varchar(255)	descripción de la cámara.
specific_info	varchar(255)	Specific_info de la cámara.

Nombre: tbl_ zona		
Descripción: Representa una zona física de la institución en la que se despliega el Sistema.		
Atributo	**Tipo**	**Descripción**
id_zona	integer	ID de la zona física
descripción	varchar(255)	Descripción
sub_zona	integer	Zona .
shortname	varchar(50)	Nombre de la zona

Nombre: tbl_rol		
Descripción: Representa los roles que pueden representar un usuario en el sistema.		
Atributo	**Tipo**	**Descripción**

Id_rol	integer(50)	Id de los roles.
nombre	varchar(50)	Nombre de los roles del sistema.

Nombre: tbl_map		
Descripción: Representa los datos de los mapas de grabación.		
Atributo	**Tipo**	**Descripción**
id_map	Integer(50)	Id de los mapas de grabación.
descripción	varchar(50)	Descripción del lugar del mapa de grabación.
data	varchar(50)	Datos de los mapas de grabación.
widht	integer(50)	Ancho de los mapas.
height	integer(50)	Altura de los mapas.

Nombre: tbl_privilegio_administracion		
Descripción: Representa los privilegios de administración.		
Atributo	**Tipo**	**Descripción**
id_entity	Integer(50)	Id de la entidad.
autonomia_grabacion	bool	Autonomía de grabación
admin_configuracion	bool	Admin configuración.
admin_usuarios	bool	Admin usuarios.

Nombre: tbl_log_seguridad		
Descripción: Representa los datos de los usuario logeados en el sistema.		
Atributo	**Tipo**	**Descripción**
id_log	Integer(50)	Id de usuario logeado.
descripción	varchar(30)	Nombre del usuario logeado.
type	varchar(30)	Tipo de logeo.

Event_date	varchar(30)	Fecha del logeo.
From_who	varchar(30)	Ip de donde se logeo el usuario.
id_entidad	integer(50)	Id de la entidad.

Nombre: tbl_log_camaras		
Descripción: Representa los datos de las cámaras que se encuentran conectadas al sistema		
Atributo	**Tipo**	**Descripción**
id_log	Integer(50)	Id de la cámara.
descripción	varchar(30)	Descripción de la cámara.
type	varchar(30)	Tipo de operación realizada por la cámara.
eventdate	time	Fecha de realizada la operación.
From_who	Varchar(20)	Ip de donde se realizó la operación.
Object_type	Varchar(30)	Tipo de objeto.
Id_objeto	Integer(50)	Id del objeto

Nombre: tbl_privilegio_camara		
Descripción: Representa un privilegio de cámara.		
Atributo	**Tipo**	**Descripción**
id_entity	Integer(30)	Id de la entidad
Id_camara	integer(30)	Id de la camara
edit	bool	Edit.
ptz	bool	Ptz.
view	bool	View.

Nombre: tbl_privilegio_grabación

Descripción: Representa un privilegio de grabación.		
Atributo	**Tipo**	**Descripción**
id_entity	Integer(30)	Id de la entidad
Id_camara	integer(30)	Id de la camara
grabar_manual	bool	grabar_manual.
calendario	bool	Calendario.
recuperación	bool	Recuperación.

Nombre: tbl_regla_grabación		
Descripción: Representa una regla de grabación		
Atributo	**Tipo**	**Descripción**
id_camara	Integer(30)	Id de la cámara que realizo la grabación.
Dia_Inicio	varchar(30)	Día de inicio de la grabación.
Dia_Fin	Varchar(30)	Día de fin de la grabación.
id_regla	Integer(30)	Id de la regla de grabación.
formato	Varchar(30)	Formato de la regla de grabación.
tipo_regla	Varchar(30)	Tipo de regla.
days	Varchar(30)	Número de días que durará la grabación.
descripción	Varchar(30)	Descripción de la regla.
nombre	Varchar(30)	Nombre de la regla de grabación.

Nombre: tbl_usuario		
Descripción: Representa un usuario del sistema.		
Atributo	**Tipo**	**Descripción**
id_usuario	Integer(30)	Id del usuario.
nombre	varchar(30)	Nombre del usuario.

password	Varchar(30)	Password del usuario.
Id_rol	Integer(30)	Id del rol del usuario.

Nombre: tbl_video		
Descripción: Representa un video del sistema.		
Atributo	**Tipo**	**Descripción**
id_video	Integer(30)	Id del video.
id_camara	integer(30)	Id de la cámara.
hard_disk_path	Varchar(30)	hard_disk_path.
fecha_inicio	Varchar(30)	Fecha de inicio.
fecha_fin	Varchar(30)	Fecha de fin.
formato	Varchar(30)	Formato del video.
lan_path	Varchar(30)	Dirección donde se encuentra el video.

4.2-MODELO DE IMPLEMENTACIÓN:

4.2.1-Diagrama de despliegue:

Un diagrama de despliegue es definido formalmente como un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes *software*, objetos, procesos (caso particular de un objeto). En general un nodo será una unidad de computación de algún tipo, desde un sensor a un *mainframe*. Las instancias de componentes *software* pueden estar unidas por relaciones de dependencia, posiblemente a interfaces (ya que un componente puede tener más de una interfaz) (14).

En esta aplicación el despliegue se encarga de describir el hardware utilizado en la implementación de la misma y las relaciones que se establecen entre sus componentes. Aquí se definen las relaciones físicas entre los distintos nodos que componen el módulo gestor. De esta manera se tiene que existe una PC

para cada módulo del proyecto los cuales no están conectados con algún otro distinto, por lo que no necesitan saber de la existencia los demás módulos sino que a través del gestor ellos obtienen todos los datos que necesiten para poder realizar sus funciones de manera normal. La vista de despliegue por tanto es la representación gráfica del sistema gestor y muestra el orden y la disposición de los componentes (que son los demás módulos del sistema: el grabador, el visor y el recuperador) en forma de nodos conectados por enlaces de comunicación.

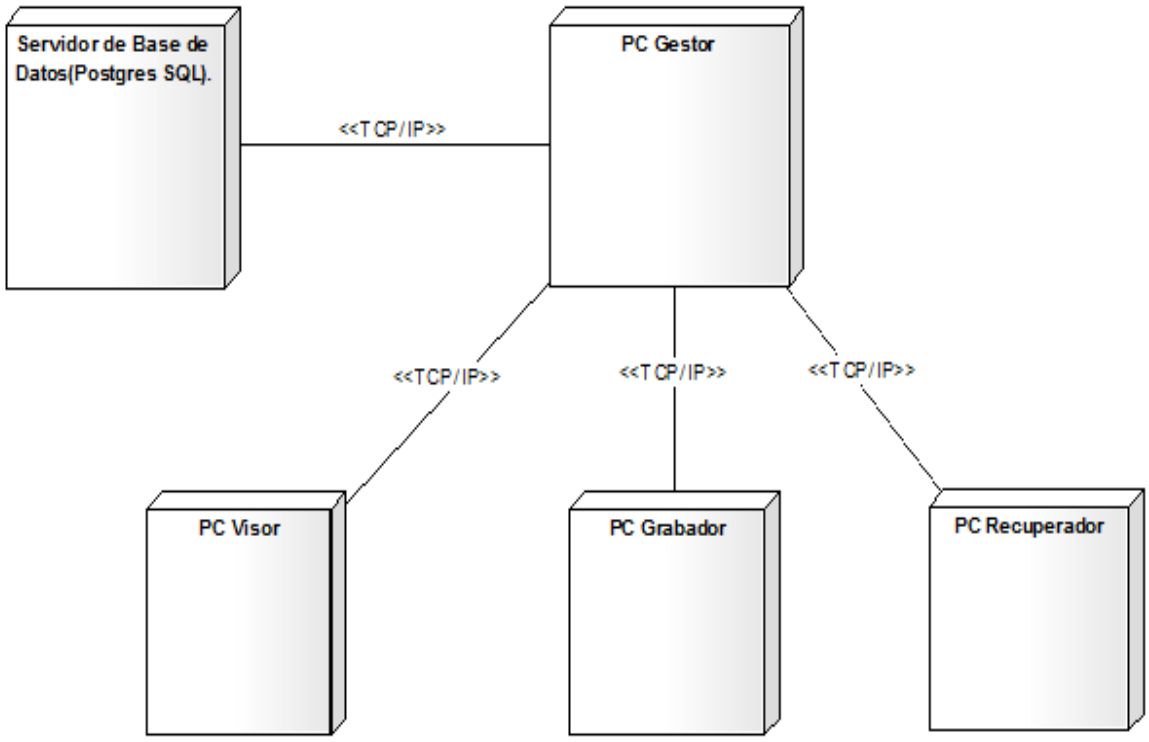

Figura 10: Diagrama de despliegue.

4.2.2-Diagrama de componentes:

El sistema cuenta con varios componentes para el manejo de la BD: siendo el más importante de ellos: Qtsql.dll que se encarga de conectarse a la misma y realizar las funcionalidades necesarias para la consultas y el manejo de datos.

Por otro lado, QtNetwork.dll realiza todo el trabajo sobre sockets, permitiendo la creación de servidores tcp y clientes para el envío y la obtención de todos los datos tanto por parte del servidor como del cliente para las pruebas parciales y posee las clases necesarias para el trabajo en la red, de forma fácil y portable.

Por último se muestra cómo queda estructurado el diagrama de componentes, desde el punto de vista de los mismos se tiene en cuenta los requisitos no funcionales relacionados a la facilidad de desarrollo, gestión de software, y las restricciones impuestas por los lenguajes de programación.

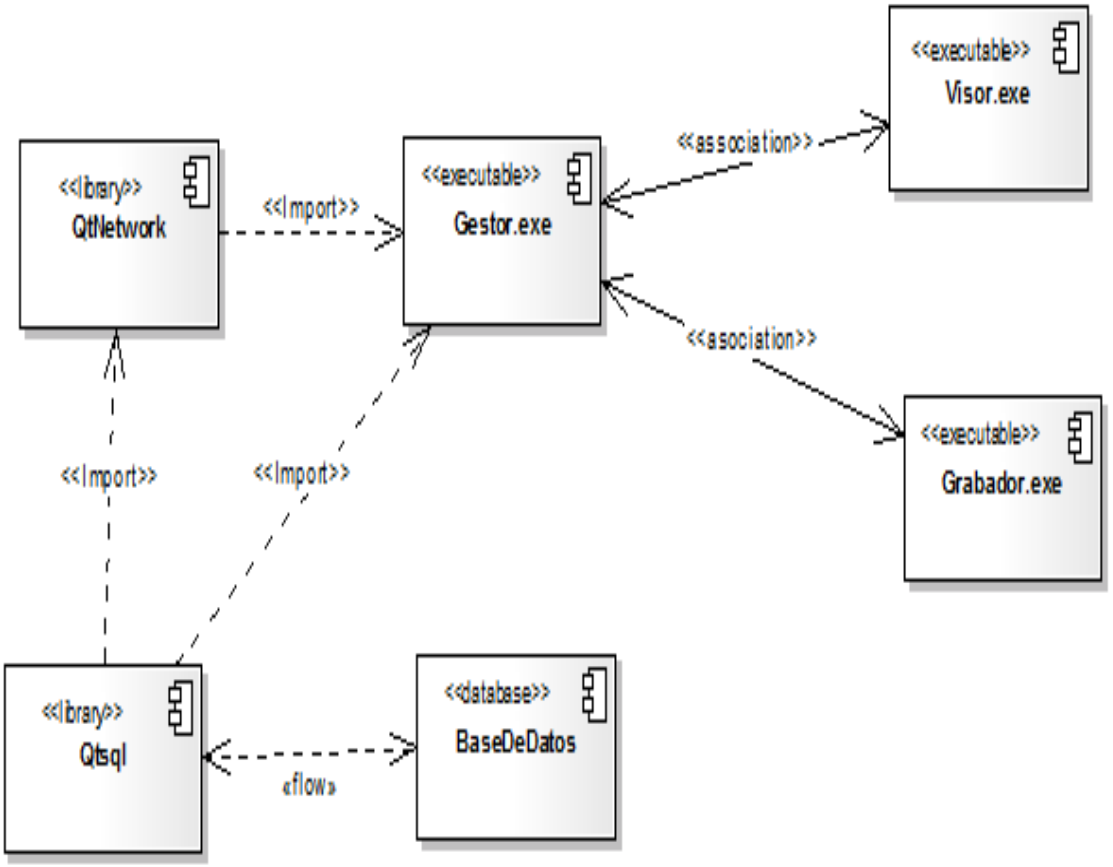

Figura 11: Diagrama de componentes.

4.3-PRUEBAS:

Esta es la última fase del ciclo de vida del desarrollo de software, **Roger Pressman** define las pruebas como:*"una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente"* (13). El flujo de trabajo de pruebas, es realizado para encontrar errores en la implementación. En el desarrollo de esta aplicación (el módulo gestor) se llevan a cabo una varios pasos en los que puede existir la

posibilidad de que surjan una serie de errores. Después de la implementación de la solución se realizaron las pruebas de caja negra, estas permiten al ingeniero de software, dado un conjunto de condiciones de entrada, probar exhaustivamente los requisitos del sistema, por lo que es necesario llevar a cabo varios casos de prueba que garanticen la calidad de la misma y que sirvan para:

- Detectar fallas o errores.
- Aumentar la calidad del producto final.
- Brindar un mayor nivel de confiabilidad en el software desarrollado.

A continuación se muestran algunos ejemplos de estos casos de prueba en este sistema ya que debido a su extensión todos no pueden estar.

1. Descripción General.

Este CU se inicia cuando el actor Sistema Administrador desea manejar la conexión a la base de datos, configurando los parámetros de la misma.

2. Secciones a probar en el Caso de Uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central.
SC 1: Manejar la conexión a la Base de Datos.	EC 1.1: Manejar la conexión a la Base de Datos con éxito.	El administrador del sistema inicializa el mismo y se muestra una ventana para entrar los datos(Databasename, password, user y DirLogs.Ejemplo:C:\Users\Fernando\Desktop\los4Proyectos\vvi.sqlite,Administrador,Administradory,C:\Users\Fernando\Desktop\los4Proyectos\Gestor\Gestor\Logs luego selecciona el botón Aceptar o Cancelar.	Módulo Gestor/ clic en "botón "Aceptar".
	EC 1.2: Manejar la conexión a la Base de Datos sin éxito.	Si la conexión esta caída o el usuario no se encuentra en el sistema de BD, se muestra un mensaje de error especificándose el mismo.	

	EC 1.3 Cancelar Manejar la conexión con la base de Datos.	El usuario selecciona el botón “Cancelar”. El sistema cancela la operación y se cierra.	Módulo Gestor/ clic en “botón “Cancelar”.

3. Descripción de variable.

No	Nombre de campo	Clasificación	Valor Nulo
1	Databasename:	Campo de texto	No
2	Username:	Campo de texto	No
3	Password:	Campo de texto	No
4	DirLogs:	Campo de texto	Si.

4. Matriz de Datos

ID	Escenario	Variable 1 Usuario	Variable 2 Contraseña	Respuesta del sistema	Resultado de la prueba.
EC 1.1	Manejar la conexión a la Base de Datos con éxito.	V/Administrador	V/Administrador.	El usuario accede al sistema con los permisos que le sean asignados según su rol.	**Satisfactorio.**
EC 1.2	Manejar la conexión a la Base de Datos sin éxito.	V/Administrador	I/administrador	El usuario no puede acceder al sistema, y se le da la opción de insertar sus datos nuevamente.	**Satisfactorio.**
EC 1.3	Cancelar Manejar la			Se cierra la aplicación.	**Satisfactorio.**

	conexión a la Base de Datos.				

1. Descripción General.

Este CU se inicia cuando el actor Sistema desea almacenar todos los logs de cada una de las operaciones que se realizan en el sistema que pasan a través de él (en especial las que conducen a un cambio del sistema) y termina cuando se registran todos los logs de las operaciones en un *.txt.

2. Condiciones de Ejecución.

La conexión a la base de datos está abierta.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central.
SC 1: Registrar logs	EC 1.1: Registrar logs con éxito.	El Sistema realiza cualquier actividad que implique un cambio en el mismo y se registra un logs de dicha operación.	MóduloGestor/Conexión.ui"/click sobre el campo de texto DirLogs.
	EC 1.2: Registrar logs sin éxito.	Si no se especifica la Dirección donde se almacenan los logs estos no son guardados.	

3. Descripción de variable.

No	Nombre de campo	Clasificación	Valor Nulo
1	DirLogs:	Campo de texto	Si

4. Matriz de Datos

ID	Escenario	Variable 1 Usuario	Respuesta del sistema	Resultado de la prueba.
EC 1.1	Registrar logs con éxito.	V/C:\tesis\los 4 Proyectos\Gestor \Gestor\Logs\file.t xt	El usuario le especifica donde debe crear el archivo para guardar los Logs de actividad.	**Satisfactorio.**
EC 1.2	Registrar logs sin éxito.	V/ C:\tesisdf\los 4 Proyectos\Gestor a\Gestor\Logs\file .txt	El sistema no puede almacenar los logs de actividad ya que la Dirección no es correcta o nula.	**Satisfactorio.**

4.4-CONCLUSIONES.

Este capítulo arrojó como resultado el despliegue que poseerá este trabajo, dicho despliegue está definido en el diagrama de despliegue y muestra los recursos del mismo y cómo estos se comunican entre sí, también se obtuvo como resultado las tablas que constituyen el modelo de datos y describen la base de datos del sistema definiendo todas las características sobre el almacenamiento físico de los mismos, se obtuvo los componentes que se encuentran en esta etapa de desarrollo, formando una visión más clara de cómo han de quedar los mismos en relación a su ubicación, sus relaciones y uso a la hora de llevar a cabo la implementación de la aplicación y por último y no menos relevante se establecieron los casos de pruebas para algunos CU del sistema, esto permite validar los errores que puedan surgir una vez terminada la implementación.

CONCLUSIONES GENERALES:

Con el desarrollo del presente trabajo de diploma, se llevaron a cabo en gran parte las tareas y objetivos trazados.

Lo que permitió arribar a las siguientes conclusiones:

- ✓ Se realizó un análisis sobre los principales sistemas de video vigilancia existente en la actualidad. Las principales limitaciones estuvieron en que este software tienen en su funcionamiento gran dependencia del fabricante. Aquí es importante destacar la capacidad del sistema de soportar variados hardware de diversos fabricantes, que es una de las limitaciones actuales de la mayoría de los sistemas en el mercado de soluciones para Video y Vigilancia. Además todos los datos que usa o genera se acogen a los estándares internacionales que rigen estos sistemas con software propietario.

- ✓ Se valoraron las principales tecnologías y herramientas actuales lo que permitió seleccionar las adecuadas para el desarrollo del sistema.

- ✓ Se implementó una aplicación que brinda servicios de gestión de datos dentro del sistema al que estén conectadas las cámaras, mediante consultas, así como la obtención de los mismos en tiempo real. Permitiendo de manejar toda la configuración referente a las cámaras IP y cumpliendo lo que se requiera para el funcionamiento de los diferentes módulos.
- ✓ Se logró que el sistema esté preparado para evolucionar hacia nuevas funcionalidades y actualizar las ya existentes.

RECOMENDACIONES:

A continuación se realizan algunas recomendaciones para la continuación de la implementación del sistema:

1.-Actualizar y mejorar las funcionalidades ya existentes así como agregar algunas nuevas que se requieran para el sistema**.**

2.-Incorporar una ayuda a la aplicación.

Bibliografía Referenciada

1. **Edmis Deivis Semanat Aldana, Leonor Verdecia Four.***Sistema de Video Vigilancia SURIA.* La Habana : s.n., 2009

2. www.innocea.com. [En línea] 2010. [Citado el: 1 de 12 de 2011.] http://www.innocea.com/Proyectos/sig/gestion_01.asp?p=1. 3.

3. **Infokrause.** www.infokrause.com. [En línea] INFOKRAUSE Y COMPAÑÍA LIMITADA . [Citado el: 8 de 03 de 2011.] http://www.camarasip.cl/que_es_una_camara_ip.htm.

4.**Fernando Echemendia Tour, Yoel Rivera Suárez.***Sistema de para la Detecciónyy Extracción de Textos en Videos Digitales.*[Citado el: 29 de mayo de 2012.]

5. **Axis Communications.** Axis Communications. *Axis Communications.* [En línea] 21 de 11 de 2008. [Citado el: 21 de 11 de 2008.] http://www.axis.com.

6. **marchnetworks.com.** electroguardian.com. [En línea] 2010. [Citado el: 1 de 12 de 2010.] http://electroguardian.com/files/Download/VS_VMS_DS_SP_12-08.pdf. 6.

7. **Soto, prof Lauro.** mitecnológico.com. [En línea] Chinavasion Wholesale Ltd. [Citado el: 20 de 3 de 2011.] http://www.mitecnologico.com/Main/EspecificacionesDeRequerimientos.

8. **Microsoft Corp.** MSDN. MSDN. [Online] Microsoft Corp. . [En línea] [Citado el: 20 de 10 de 2011.] http://www.msdn.com.

9.*qt.nokia.com.* [En línea] [Citado el: 15 de diciembre de 2011.] ttp://www.qt.nokia.com.

10. **sparxsystems.com.au.** entrebits.com. [En línea] 8 de 4 de 2008. [Citado el: 23 de 3 de 2011.] http://www.entrebits.com/descargas/productividad-y-negocios/empresas-y-negocios-especificos/enterprise-architect/.

11. PostgreSQL8.4 ya disponible. *Portal de Astra NTi.* [En línea] [Citado el: 25 de November de 2011.] http://www.astra.es/noticias/postgresql-8-4-ya-disponible.

12. **Expósito, Erly Delgado.** monografias.com. [En línea] 2010. [Citado el: 1 de 12 de 2010.] http://www.monografias.com/trabajos60/metodologias-desarrollo-software/metodologias-desarrollo-software.shtml. 7.

13. **Pressman, Roger.S.***Ingeniería de Software un enfoque práctico.* Mexico : Mcgraw Hill - Mexico, 2010. 6071503140.

14.Virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc. [En línea] [Citado el: 29 de 3 de 2011.]

15.**Rayner Pupo Gómez.** Estación de Monitorización para Video Vigilancia.[Citado el: 5 de junio de 2012.]

Bibliografía

1. **Infokrause.** www.infokrause.com. [En línea] INFOKRAUSE Y COMPAÑÍA LIMITADA . [Citado el: 8 de 03 de 2011.] http://www.camarasip.cl/que_es_una_camara_ip.htm.

2. www.axis.com. [En línea] Axis comunication, 2010. [Citado el: 1 de 12 de 2010.] http://www.axis.com/es/products/cam_station_software/. 4.

3. **marchnetworks.com.** electroguardian.com. [En línea] 2010. [Citado el: 1 de 12 de 2010.] http://electroguardian.com/files/Download/VS_VMS_DS_SP_12-08.pdf. 6.

4. **Lagos, Camilo Zambrano.** www.trickyweb.cl. [En línea] Creative Commons Attribution-Noncommercial-Share Alike 2.0., 2 de 9 de 2009. [Citado el: 15 de 3 de 2011.] http://www.trickyweb.cl/2009/09/02/c-c-c-cual-es-la-diferencia/?wscr=1024x768.

5. **IntelegoSoft.** IntelegoSoft.com. [En línea] IntelegoSoft. [Citado el: 5 de 4 de 2011.] http://www.intelegosoft.com/esp/ea/index.asp.

6. **Expósito, Erly Delgado.** monografias.com. [En línea] 2010. [Citado el: 1 de 12 de 2010.] http://www.monografias.com/trabajos60/metodologias-desarrollo-software/metodologias-desarrollo-software.shtml. 7.

7. **Solís., Manuel Calero.** willydev.net. [En línea] [Citado el: 23 de 3 de 2011.] http://www.willydev.net/descargas/prev/ExplicaXp.pdf.

8. **Soto, prof Lauro.** mitecnológico.com. [En línea] Chinavasion Wholesale Ltd. [Citado el: 20 de 3 de 2011.] http://www.mitecnologico.com/Main/EspecificacionesDeRequerimientos.

9. **Systems, Bosch Security.** esource.boschsecurity.com. [En línea] 2010. [Citado el: 1 de 12 de 2011.] http://resource.boschsecurity.com/documents/BoschVideoManag_DataSheet_esES_T3632348427.pdf. 5.

10. **Garcerant, Ivan.** Tecnología y Sinergix. [En línea] 10 de 7 de 2008. [Citado el: 25 de 2 de 2011.] http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/.

11. **Martínez, Gerardo Moreno.** monografías.com. [En línea] [Citado el: 24 de 2 de 2011.] http://www.monografias.com/trabajos5/andi/andi.shtml.

12. dcc.uchile.cl. [En línea] [Citado el: 24 de 02 de 2011.] http://www.dcc.uchile.cl/~psalinas/uml/modelo.html.

13. **Alvarez Zayas, Carlos.***Metodología de la Investigación.* Ciudad de la Habana : ECIMED, 1989.

14. **Morales, Byron.** SCribd. *Metodologías de Desarrollo de Software.* [En línea] 9 de febrero de 2011. [Citado el: 2 de marzo de 2011.] http://es.scribd.com/doc/13349706/Justificacion-metodologia-XP.

15. **Garcerant, Iván.** synergix.wordpress.com. [En línea] Tecnología y Synergix, 10 de 7 de 2008. [Citado el: 23 de 03 de 2011.] http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/.

16. **sparxsystems.com.au.** entrebits.com. [En línea] 8 de 4 de 2008. [Citado el: 23 de 3 de 2011.] http://www.entrebits.com/descargas/productividad-y-negocios/empresas-y-negocios-especificos/enterprise-architect/.

17. **Microsoft Corp.** MSDN. MSDN. [Online] Microsoft Corp. . [En línea] [Citado el: 20 de 10 de 2011.] http://www.msdn.com..

18. **Jiménez, Javier Alonso Albusac.***Vigilancia Inteligente: Modelado de Entornos Reales* l 2008.

19. Rational Software Rational Software IBM online. *Rational Software Rational Software IBM online.* [En línea] [Citado el: 24 de 11 de 2010.]

20. **Edmis Deivis Semanat Aldana, Leonor Verdecia Four.***Sistema de Video Vigilancia SURIA.* La Habana : s.n., 2009.

21.**Fernando Echemendia Tour, Yoel Rivera Suárez.***Sistema de para la Detecciónу Extracción de Textos en Videos Digitales.*[Citado el: 29 de mayo de 2012.]

23. VIVOTEK - Cámaras de red. *VIVOTEK - Cámaras de red.* [En línea] [Citado el: 2 de 12 de 2011.] http://www.vivotek.com/.../network_cameras.php?newlang.

24. Información general de .NET Framework Remoting. *Visual Studio.* [En línea] [Citado el: 25 de November de 2011.] http://msdn.microsoft.com/es-es/library/kwdt6w2k(VS.80).aspx .

25. PostgreSQL8.4 ya disponible. *Portal de Astra NTi.* [En línea] [Citado el: 25 de November de 2011.] http://www.astra.es/noticias/postgresql-8-3-ya-disponible.

26. **Axis Communications.** Axis Communications. *Axis Communications.* [En línea] 21 de 11 de 2008. [Citado el: 21 de 11 de 2008.] http://www.axis.com.

27. **Pressman, Roger.S.***Ingeniería de Software un enfoque práctico.* Mexico : Mcgraw Hill - Mexico, 2010. 6071503140.

28. **Cabanes, Nacho.** Diccionario de informática. *Diccionario de informática.* [En línea] 04 de 02 de 2007. [Citado el: 04 de 03 de 2011.] http://www.nachocabanes.com/diccio/ndic.php.

29. **Molina, Rafael.***Introducción al procesamiento y análisis de imágenes digitales.* Granada : s.n., 1998.

30. IngenieroSoftware. [En línea] [Citado el: 10 de 12 de 2011.] http://www.ingenierosoftware.com/analisisydiseno/uml.php.

31. **Molpeceres, Alberto.** Procesos de Desarrollo RUP, XP . [En línea] 15 de 12 de 2002. [Citado el: 9 de 2 de 2011.] http://www.willydev.net/descargas/articulos/general/cualxpfddrup.PDF.

32. **Microsoft.** Office Online. *Office Online.* [En línea] 2008. [Citado el: 10 de 3 de 2011.] http://office.microsoft.com/es-hn/help/CH010001183082.aspx.

33. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.***El Proceso Unificado de Desarrollo.* 2000.[Citado el :23 de marzo 2011]

34. **Schmuller, Joseph.***Aprendiendo UML en 24 horas.* 2000.

35. www.innocea.com. [En línea] 2010. [Citado el: 1 de 12 de 2011.] http://www.innocea.com/Proyectos/sig/gestion_01.asp?p=1. 3.

36.Virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc. [En línea] [Citado el: 29 de 3 de 2011.]

37.*qt.nokia.com.* [En línea] [Citado el: 15 de diciembre de 2011.]

ttp://www.qt.nokia.com.

38.**Rayner Pupo Gómez.** Estación de Monitorización para Video Vigilancia.[Citado el: 5 de junio de 2012.]

Glosario de términos:

Actor del negocio.	Los actores del negocio como aquellos con los que el sistema interactúa.
CASE	Ingeniería de software asistida por computadora, por sus siglas en idioma Inglés.
Clase.	Estructura de datos empleada para crear objetos.
Casos de uso.	Los CU se identifican a partir de los RF, los mismos llevan a cabo una serie de acciones que responden a un evento disparado por algún actor sobre el sistema.
Diagrama de CU.	Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas.
IDE.	Entorno Integrado de Desarrollo (Siglas en inglés).
PTZ.	Es un acrónimo de pan-tilt-zoom, las cámaras PTZ pueden moverse horizontalmente, verticalmente y acercarse o alejarse de un área o un objeto de forma manual o automática.
Requisito funcional.	Capacidad o condición que el sistema debe cumplir (15).
Requisito no funcional.	Propiedad o cualidad que el sistema debe tener (15).

Sockets.	Designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.
UML	El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema software (13).
Video-Vigilancia.	Mecanismo de supervisión mediante el uso de cámaras.
Visor.	Componente gráfico que representa la señal emitida por una cámara mediante el uso de un render determinado.