UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 5

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS.

Título:

Desarrollo de un componente para visualizar los flujos de videos de una cámara IP en el Módulo HMI del SCADA GALBA.

Autor: Eduardo Grabiel Luzúa Gómez.

Tutor(es): Ing. Yuremis Mengana Claro.

Co-tutor(es): Ing. Andris Villalón De la Cruz.

Asesor: Ing. Ridel Oscar García Mora.

DECLARACIÓN DE AUTORÍA.

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

_____

Firma del Autor: Eduardo Grabiel Luzúa Gómez.

_____

Firma del Tutor: Ing. Yuremis Mengana Claro.

_____

Firma del Co-Tutor: Ing. Andris Villalón De la Cruz.

Dedicatoria

Este trabajo está dedicado especialmente a mi mamá quien ha sufrido y llorado cada una de las trastadas de su hijo cabezón.

A papá, en donde quiera que estés, sé que te sientes orgulloso de tu hijo ingeniero.

A mi hermano, que siempre está a mi lado en las buenas y en las malas.

A mis abuelos Nancy y Yayo mis segundos padres.

A toda mi familia, que siempre ha estado dándome todo el apoyo que he necesitado.

Agradecimientos

A mi tía Rosita, gracias por tus consejos invaluables, has sido mi guía en muchas de las decisiones difíciles que tomado.

A Rafael, por ser el que encendió mi chispa de informático.

A tía Teresa por brindarme su apoyo en todo momento.

A mi abuela Edice por estar siempre dispuesta a ayudarme.

A mis primos por estar siempre a mi lado.

A mi hermano Mera, compañero de viajes, lograr este sueño en parte te lo debo a ti.

A Yindra por estar siempre a mi lado, en las buenas y en las malas.

A José Ernesto, hermano de toda la vida y compañero de parrandas.

A mis amigos de toda la vida, aunque no estén presentes conmigo, sé que me llevan en el pensamiento.

A los colegas del SCADA que siempre me brindaron todo el apoyo que necesite.

A mis tutores, especialmente a Andris sin su ayuda todo hubiera sido más difícil.

A todos los que han hecho posible que este día se haga realidad.

Resumen

Los sistemas de video vigilancia a través de cámaras IP han ganado gran popularidad en la actualidad, fortaleciendo los sistemas de seguridad existentes, esto se debe a que pueden ser instalados en diferentes lugares y en diversas condiciones ambientales. El Centro de Informática Industrial (CEDIN), perteneciente a la Universidad de las Ciencias Informáticas (UCI), mantiene un contrato con la Empresa Socialista de Capital Mixto Guardián del Alba (ESCMGA), el cual consiste en el desarrollo de un sistema de supervisión, control y adquisición de datos (SCADA). Este sistema está compuesto por varios módulos entre los que se encuentra el Interfaz Hombre-Máquina (HMI).

El módulo HMI permite configurar, supervisar y controlar los procesos industriales. Actualmente tiene como limitante la no existencia de un mecanismo que permita supervisar de forma eficiente el proceso de quemado de gas natural y las zonas de acceso restringido. Este documento describe el desarrollo de un componente para la visualización de videos en tiempo real en el módulo HMI del SCADA GALBA, sirviendo como base para la incorporación de nuevas funcionalidades a dicho módulo, además de brindar nuevas características como la visualización de flujos de videos para monitorizar el proceso de quemado de gas resultante de la extracción de petróleo, así como la visualización de las zonas de acceso restringido. Para el desarrollo del componente se empleó la metodología de desarrollo de *software Extreme Programming* (XP), el marco de trabajo Qt, el entorno de desarrollo integrado *QtCreator* y la biblioteca *libVLC* para el trabajo con los flujos de videos.

Palabras clave: Cámara, componente, flujos de video, video vigilancia.

ÍNDICE

ÍNDICE DE TABLAS

INTRODUCCIÓN

El comienzo de la Revolución Industrial en el siglo XVIII supuso un punto de inflexión en la historia de la humanidad, el aumento de las fuerzas productivas dio comienzo a un desarrollo más eficiente en todas las esferas de la industria, así como todos los sub-procesos involucrados dentro de la misma; con la Revolución Tecnológica del pasado siglo se incorporaron nuevas ciencias tales como la informática y la robótica llevando el proceso automatizado a un nivel superior. Aparecen los primeros sistemas para la supervisión, control y adquisición de datos (SCADA) en los procesos industriales consolidando el dominio sobre la producción de las plantas industriales y favoreciendo el desarrollo de las mismas.

Los sistemas SCADA, son sistemas informáticos que se encargan, como su nombre indica, de la supervisión y el control del flujo de datos en cualquier planta industrial a distancia y en tiempo real, su principal uso radica en las telecomunicaciones, sistemas hidráulicos, eléctricos, el sector petrolero, entre otros sectores de gran importancia. El hecho de que estos sistemas se desplieguen en plantas industriales, en muchos casos de importancia vital para la industria (ej. la petrolera), reviste también la necesidad de incorporar junto con los mismos, sistemas para apoyar la seguridad de dichas instalaciones. Dentro de los más populares en la actualidad se encuentran los sistemas de video vigilancia.

Los Sistemas de video vigilancia aparecidos en la década del 60 brindaron una nueva tecnología para el reforzamiento de los sistemas de seguridad en las plantas industriales, permitiendo la observación en tiempo real de todo lo ocurrido en el entorno de una instalación. La llegada de Internet supuso un salto de avance en el concepto que se tenía hasta ese momento del proceso de video vigilancia, nuevos elementos comenzaron a formar parte de ese eslabón de seguridad dentro de los cuales las cámaras IP[1], fusionaron todos los avances logrados hasta el momento en dos ramas del desarrollo humano, las comunicaciones y la informática.

La cámara IP es una cámara de red que combina los beneficios propios de los Circuitos Cerrados de Televisión (CCTV) tradicionales y las ventajas digitales de las redes de comunicación IP, permitiendo la supervisión local y/o remota de imágenes y audio, así como el tratamiento digital de las imágenes, incorpora su propio miniordenador permitiendo enviar videos y observar en tiempo real lo que está ocurriendo, entre otras facilidades.

[1] IP: Protocolo de Internet por sus siglas en inglés.

En la Universidad de la Ciencias Informáticas, está establecido el Centro de Informática Industrial (CEDIN), el cual, entre sus múltiples líneas de desarrollo cuenta con el sistema SCADA Guardián del ALBA (GALBA), el mismo se enfoca en la automatización de los procesos industriales dentro del sector petrolero venezolano, al estar desplegado en varias instalaciones de la empresa estatal Petróleos de Venezuela (PDVSA).

Este sistema tiene una importancia vital en el desenvolvimiento del sector petrolero en Venezuela, por lo cual, es imprescindible incorporar cada cierto tiempo nuevas características al módulo de seguridad para facilitar el desarrollo normal de la infraestructura productiva. La quema del gas acompañante, resultado de la extracción de petróleo, es uno de los momentos en los cuales debe existir una constante supervisión por parte de los trabajadores de la planta para evitar accidentes que atenten contra la salud de los trabajadores, la infraestructura productiva y la población en general.

El personal responsable en el proceso de quemado del gas, ha manifestado la necesidad de supervisar dicho evento, debido a que aún no se cuenta con el equipamiento necesario para supervisar dicho proceso. Actualmente se cuenta con un simple sistema de video vigilancia, el cual no está integrado al sistema SCADA, teniendo como inconveniente que el operador debe estar observando varios monitores ubicados en distintas posiciones, ante esta situación puede darse el caso de que al ocurrir una señal de alarma el operador no se encuentre atento al monitor que notifique la anomalía en el sistema, también se desea que el operador cuente con la posibilidad de guardar secuencias de video para en caso de ser requerido estos puedan ser consultados. Por otro lado, existen áreas dentro de las instalaciones petroleras que tienen gran importancia, ya sea por su ubicación geográfica o por los procesos industriales que en ellas se desarrollan. Muchas de estas áreas no pueden ser accedidas por el personal responsable, puesto que el acceso a las mismas resulta difícil y engorroso. Teniendo en cuenta que estas zonas son de gran importancia económica, y que un accidente producido en cualquiera de ellas, podría atentar contra la seguridad de las instalaciones petroleras y la seguridad del personal que allí trabaja, se hace evidente la necesidad de crear un mecanismo que permita tener vigiladas constantemente y de forma eficiente estas zonas.

Ante la problemática planteada se define como **problema a resolver:** ¿Cómo supervisar a través del SCADA GALBA, de manera remota y en tiempo real, las diferentes áreas en las instalaciones petroleras? Por lo que se delimita como **objeto de estudio:** los procesos de supervisión a través de cámaras IP. Para dar solución al problema de la investigación científica se define como **objetivo general:** desarrollar un

componente que permita supervisar de manera remota y en tiempo real, las diferentes áreas en las instalaciones petroleras. Enmarcando en el **campo de acción:** la supervisión de manera remota y en tiempo real de las diferentes áreas de las instalaciones petroleras, desde el SCADA Guardián del Alba. Para dar cumplimiento al objetivo anteriormente planteado se definen las siguientes **tareas de investigación:**

- Estudio y análisis de soluciones existentes en el campo de la video vigilancia.
- Estudio acerca de la arquitectura del HMI SCADA.
- Realizar un resumen de los sistemas de video vigilancia, con el objetivo de elegir un sistema que se ajuste a las necesidades del proyecto.
- Selección de tecnologías, herramientas y medios para el desarrollo de una aplicación que permita la visualización de flujos de videos de cámaras IP.
- Implementación de un componente para la visualización de flujos de videos de cámaras IP en el módulo HMI de SCADA GALBA.
- Realizar pruebas para comprobar el correcto funcionamiento del componente.

Métodos Teóricos

Análisis - Síntesis: este método permite buscar la esencia de los fenómenos, los rasgos que lo caracterizan y los distinguen. Su objetivo en una investigación es analizar las teorías, documentos, etc., permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio. Permitió a partir de la información y documentación estudiada sobre el tema, en diferentes fuentes, la obtención de un conocimiento general para apoyar el desarrollo de la investigación.

Histórico - Lógico: permite estudiar de forma analítica la trayectoria histórica real de los fenómenos, su evolución y desarrollo. Su objetivo en una investigación es constatar teóricamente cómo ha evolucionado un determinado fenómeno en un período de tiempo, en toda su trayectoria o en un fragmento temporal de la lógica de su desarrollo. Permitió realizar un análisis profundo sobre la evolución y comportamiento de los sistemas que permiten la visualización de flujos de videos de cámaras IP durante las últimas décadas.

Modelación: es una reproducción simplificada de la realidad, cumple una función heurística, que permite descubrir y estudiar nuevas relaciones y cualidades del objeto de estudio. La modelación es justamente el proceso mediante el cual creamos modelos con vistas a investigar la realidad. Permitió comprender mejor el funcionamiento que debía tener el componente, así como una mejor conformación mediante la

realización de los diagramas necesarios para una correcta construcción del mismo.

Métodos Empíricos

Entrevista: es una conversación planificada para obtener información. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos o sobre características personales del entrevistado y puede influir en determinados aspectos de la conducta humana por lo que es importante una buena comunicación. Permitió identificar la situación problemática y la obtención de los requisitos funcionales del sistema a partir de una entrevista realizada de forma intencional al personal involucrado con dicho tema, específicamente al personal del proyecto video vigilancia SURIA del centro GEYSED, al arquitecto del SCADA GALBA del centro CEDIN, además de otros compañeros vinculados al proyecto de supervisión por cámaras IP de la empresa XETID.

El presente documento se encuentra estructurado en tres capítulos, los cuales se describen a continuación:

CAPÍTULO 1. Fundamentación teórica: En este capítulo se presentan los principales conceptos que se deben tener en cuenta para lograr un mejor entendimiento del componente que se desea desarrollar, se hace un estudio de la evolución históricas de los sistemas de video vigilancia, se realiza un estudio de los casos homólogos existentes, además de la selección de las herramientas y tecnologías a utilizar durante el desarrollo del componente.

CAPÍTULO 2. Análisis y diseño del sistema: Durante este capítulo se realiza una propuesta de solución del sistema que se pretende desarrollar, se elaboran las Historias de Usuarios para identificar las funcionalidades con las que debe cumplir el componente, así como los principales requerimientos no funcionales, se hace el diseño de la solución propuesta mediante las tarjetas CRC, diagramas de clases y diagramas de paquetes, además de presentar la arquitectura seleccionada para el desarrollo del sistema.

CAPÍTULO 3. Implementación y prueba: Se realiza la implementación del componente de acuerdo a las funcionalidades y clases diseñadas previamente. Se muestra un diagrama de componentes, el cual permite tener una mejor idea de cómo funciona la integración entre los componentes del sistema, además se muestra una vista de cómo quedará desplegado el sistema una vez desarrollado y se realizan las pruebas de aceptación, las cuales validan el correcto funcionamiento del componente.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se abordan los conceptos y temas teóricos relacionados con el problema planteado. Se realizará un análisis de la evolución y tendencias en el mundo sobre la vigilancia a través de videocámaras desde sus inicios hasta la actualidad, así como el desarrollo de las cámaras IP, teniendo en cuenta las diferentes alternativas existentes en el mercado. En el último segmento se define la metodología de desarrollo de *software* así como la herramienta de diseño y modelado a utilizar en el desarrollo del componente.

1.2 Definiciones generales

A continuación se describen los principales conceptos relacionados con la concepción del sistema, estas definiciones servirán como punto de partida para el desarrollo del presente trabajo.

Video

Existen varias definiciones para referirse al termino de video, según el sitio web ***www.definiciones.de*** “es un sistema de grabación y reproducción de imágenes, que pueden estar acompañadas de sonidos y que se realiza a través de una cinta magnética. (…) consiste en la captura de una serie de fotografías (en este contexto llamadas “fotogramas”) que luego se muestran en secuencia y a gran velocidad para reconstruir la escena original.” (1)

Por otro lado, según Alberto del Bimbo en su libro “Bases de Datos de Imágenes y Videos” el vídeo “está compuesto por fotogramas, que son cada una de las imágenes individuales de una secuencia o animación, por tomas, que son las secuencias de imágenes realizadas en un mismo plano y por escenas, que representan un conjunto de tomas relacionadas entre sí.” (2)

Como se puede apreciar en ambas bibliografías consultadas existen varias coincidencias en cuanto al término de video, por lo cual podemos llegar a la siguiente conclusión: un video se puede definir como una secuencia de imágenes denominadas fotogramas, las cuales mostradas en secuencia y a grandes velocidades dan la sensación de movimiento, los mismos son captados por medios electrónicos digitales o analógicos.

Video vigilancia

Durante el desarrollo de la investigación se estudiaron diferentes conceptos de video vigilancia dentro de

los cuales se destacan los siguientes: “es toda aquella actividad que supone la colocación de una cámara de grabación, fija o móvil, con la finalidad de reforzar la seguridad de una instalación o de las personas, asegurando el correcto desempeño de las tareas en el entorno laboral, siendo de gran utilidad en diversos ámbitos” (3), por otro lado la Real Academia Española plantea que la video vigilancia es la acción de realizar “vigilancia a través de un sistema de cámaras, fijas o móviles”. (4)

Por lo tanto podemos deducir que video vigilancia hace referencia a la supervisión local o remota de imágenes de video captadas por cámaras (y sonido si las cámaras integran micrófonos). Las imágenes pueden ser emitidas en tiempo real o grabadas, permitiendo entre otras utilidades establecer pautas de comportamiento que supongan intervención policial o afecten a la seguridad ciudadana.

Cámara IP

Una cámara IP (o cámara de red) puede describirse como una cámara y un ordenador combinados para formar una única unidad inteligente. Captura y envía vídeos en directo a través de una red IP, como una Red Área Local o LAN por sus siglas en inglés, Intranet o Internet, permite a los usuarios ver o gestionar la cámara con un navegador Web estándar o con *software* de gestión de vídeo en cualquier equipo local o remoto conectado a una red. Permite a usuarios autorizados desde distintas ubicaciones acceder simultáneamente a las imágenes captadas por la misma cámara IP de red. (5)

En otra acepción podemos encontrar 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 el protocolo TCP/IP[2] sin necesidad de una computadora. Dentro de las cámaras IP, además de la cámara en sí, lleva incorporado un microordenador que es lo que permite cumplir con las funciones de comunicación anteriormente descritas.” (6)

Por lo expuesto anteriormente podemos llegar a la conclusión de que una cámara IP es aquel dispositivo que permite capturar y emitir flujos de videos, se conecta directamente a la red empleando el protocolo TCP/IP, al cual se le asigna una dirección IP de forma tal que se pueda acceder a sus funciones desde la red, para ser empleadas por aplicaciones capaces de detectarla y emplearla.

[2] Protocolo de Control de Transmisión/Protocolo de Internet por sus siglas en inglés, es un protocolo empleado en la entrega de datos a través de la red, garantiza que los paquetes de información sean entregados en el mismo orden en el cual fueron enviados.

1.3 Evolución de los sistemas de video vigilancia

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

Los sistemas de video vigilancia han evolucionado atravesando diferentes etapas en las últimas décadas. La primera generación de sistemas de video vigilancia empleaba señal y transmisión analógica, este tipo de sistema utilizaba cables coaxiales de 75 Ohm[3] para interconectar las cámaras, logrando visualizar imágenes en tiempo real en los monitores. El sistema podía contar con dos variantes, un monitor y un *switch*[4] para cambiar a la cámara deseada, o con monitores capaces de aceptar múltiples fuentes de video en ventanas separadas.

Los sistemas analógicos empleaban amplificadores de baja potencia y cables coaxiales para ver de forma remota las imágenes de las cámaras de seguridad. Las cámaras y los monitores se encontraban interconectados mediante una red de conmutación y distribución de video, compleja y cara, que direccionaba las imágenes de cada cámara hacia un monitor. La tecnología CCTV tenía demasiadas limitaciones y defectos. El video analógico solo se podía distribuir en una red local de cables coaxiales.

La segunda generación de sistemas de vigilancia se basaba, principalmente, en métodos de procesamiento y comunicación híbridos analógico-digitales, o completamente digitales. Estos sistemas aprovechaban la flexibilidad ofrecida por los primeros algoritmos de procesamiento de video, los que permitían centrar la atención del operador en un grupo de situaciones de interés, además de las facilidades proporcionadas por los primeros métodos de compresión digital para aprovechar el ancho de banda de la transmisión. En esta segunda generación es que se comienza a utilizar por primera vez *Software* para visualizar y manejar las cámaras de vigilancia, en este momento solo gestionaban información referente a las cámaras y usuarios, brindaban la funcionalidad de visualizar y manipular las cámaras que permitían las funciones PTZ[5]. Las cámaras IP PTZ pueden ser incorporadas actualmente en la infraestructura de red existente en los edificios. Estos sistemas explotan los beneficios de esta infraestructura a diferencia del cable coaxial. Con este sistema el costo de una estación de monitorización central se reduce, los movimientos, adiciones y cambios son más fáciles, ya que las cámaras pueden

[3] Unidad de medida que se refiere a la resistencia eléctrica.
[4] En este caso se refiere al interruptor que era utilizado para seleccionar la cámara a visualizar en el monitor.
[5] Acrónimo de Pan-Tilt-Zoom, las cámaras PTZ pueden moverse horizontal y verticalmente, acercarse o alejarse según sea necesario para operador.

instalarse en cualquier lugar siempre y cuando exista una toma de red. El cableado viaja hacia un multiplexor que soporta conectores RJ45, denominación que también se le da a los conectores comunes para los cables de red. Los videos digitales se graban en unidades de discos duros de la misma forma en que un archivo se almacena en una computadora. Esto permite obtener monitorización descentralizada, mejor calidad de imagen y mayor conservación de las grabaciones. Las transmisiones digitales pueden almacenarse sin la necesidad de intervención humana o cambio de cintas. Los tiempos de grabación son mayores gracias a los algoritmos de compresión dentro de los dispositivos, estas grabaciones son accesibles de forma instantánea además de poder ser visualizadas desde cualquier lugar del mundo a través de Internet.

El surgimiento del concepto de video vigilancia IP rompió con las barreras de la video vigilancia tradicional solucionando muchas de las desventajas que acarreaba el uso de los CCTV tradicionales, se eleva el alcance de las infraestructuras de seguridad por video, minimizando los costos de los mismos al aprovecharse las redes LAN/WAN[6], las redes telefónicas, inalámbricas e Internet como vía de comunicación y transmisión de datos.

Actualmente, se está produciendo una migración de los sistemas de vídeo clásicos a los sistemas de tercera generación. Los sistemas de tercera generación aprovechan el progreso de las redes de ordenadores de bajo costo y alto rendimiento, las comunicaciones multimedia fijas y móviles. La investigación en este campo trabaja en técnicas distribuidas de procesamiento de video, para el procesado de secuencias de video en tiempo real con la intención de conseguir sistemas robustos de transmisión de imagen, procesamiento de imagen en color, generación de alarmas basada en eventos, reconstrucción de secuencias a partir de modelos, segmentación y análisis en tiempo real a las secuencias de imágenes en dos dimensiones, identificación y seguimiento de múltiples objetos en escenas complejas, reconocimiento de comportamientos humanos, etc. Se prevé que todos estos trabajos proporcionen a las aplicaciones de vigilancia resultados cada vez más interesantes, gracias a la disponibilidad de una potencia de cálculo muy elevada y con precios aceptables.

Las cámaras IP, desde los primeros antecedentes han avanzado a ritmo vertiginoso hasta la actualidad. Si en un principio solo se contaba con simples cámaras web conectadas a una red, hoy se cuenta con una amplia gama para todo tipo de situaciones, desde las de altas prestaciones hasta las más modestas, fijas

[6] WAN: Red de Área Amplia por sus siglas en inglés, usualmente se refiere a varias LAN interconectadas entre sí.

y móviles, con visión infrarroja, y un sinfín de características adicionales. Dentro de los principales fabricantes podemos encontrar *Axis*, pionero en el empleo de dicha tecnología, *Vivotek*, *AVTECH*, *Foscam* y muchísimos otros fabricantes.

1.4 Estudio de los casos homólogos

A nivel mundial los sistemas de video vigilancia gozan de gran popularidad, a continuación se enunciará algunos de los casos prácticos empleados a nivel mundial:

Vijeo Citect V7.2

Software desarrollado por la empresa multinacional francesa *Schneider Electric*, especializado en la supervisión de procesos industriales automatizados, dentro de las características de este sistema se encuentran:

1. Es compatible con todas las versiones del sistema operativo Windows desde XP hasta las actuales (Vista, 7, 8), así como las versiones para servidores de las mismas 2003 y 2008.
2. Dispone de la capacidad de colocar entradas de video para cámaras IP en las pantallas del SCADA.
3. Parámetros adicionales a las variables que le dará a los usuarios información completa de los valores en tiempo real (calidad y fecha/hora), para cada variable.

Beneficios que proporciona el sistema:

- Limita la necesidad de visitas en sitio a incidentes de seguridad o de proceso, contribuyendo a una administración optimizada del personal de campo, reducción de costos de viaje, etc.
- Mejora la efectividad de la intervención de los operadores a través de información detallada que les permitirá tomar decisiones de negocio y por consecuencia reducir riesgos potenciales al personal en condiciones peligrosas.
- Acelera los tiempos de respuesta ante una emergencia, reduciendo la interrupción de operaciones.

Desventajas:

- Es un *software* propietario por el cual se deben pagar costosas licencias.
- Está reducido al uso exclusivo de cámaras IP del fabricante *PELCO*.
- No es soportado por sistemas operativos de código abierto, ej. GNU/Linux.

Ingesys IT

Sistema desarrollado por la transnacional *Ingeteam* con sede en el parque industrial de Vizcaya, España, permite la integración de un sistema de seguridad distribuido en combinación con las funcionalidades propias de un SCADA, dentro de los beneficios del mismo podemos destacar:

1. Ofrece conectividad con todo tipo de dispositivos de seguridad, cámaras IP, PLC[7], dispositivos de audio IP bidireccional, grabadores, mecanismos para el control de acceso, etc.
2. Permite coordinar la gestión de datos de los procesos con Audio y Video IP en tiempo real con una eficiente y completa administración de la información de diagnóstico de la red instalada.
3. Eficiente coordinación con el Módulo de Gestión de Alarmas.

Desventajas:

- El módulo de video vigilancia está completamente integrado al *software* de supervisión *Ingesys* imposibilitando su uso en otros sistemas de supervisión.
- Es un *software* privativo.
- Cuenta con soporte solo para el Sistema Operativo Windows.

XILEMA SURIA

Esta solución desarrollada por el centro GEYSED en la Universidad de las Ciencias Informáticas, destaca entre otras ventajas:

- Dispone de la capacidad de operar con cámaras IP de distintos fabricantes.
- Permitir el análisis en tiempo real de los flujos de video.
- Sistema flexible y escalable para conseguir una mayor adecuación a las necesidades de los clientes.

Desventajas:

- No ofrece soporte para la integración con sistemas HMI SCADA.
- Algunos de sus principales módulos todavía se encuentran en desarrollo.

El estudio realizado a los sistemas anteriormente descritos ha aportado una visión más clara de la línea a

[7] Controlador Lógico Programable: Una especie de micro-computadora usada para automatizar procesos industriales.

seguir durante el desarrollo del presente trabajo. Estos sistemas cuentan con un conjunto de características que no son idóneas para ser utilizadas como solución a la problemática planteada, entre las que se destacan:

- ✓ Son soluciones de código privativo por lo cual deben pagarse costosas licencias.
- ✓ Están integradas a sistemas cerrados por lo cual no es posible se incorporación a otros sistemas SCADA.
- ✓ Están desarrollados específicamente para algunas versiones del sistema operativo Windows.
- ✓ Las cámaras IP utilizadas por estos sistemas limitan su uso a fabricantes específicos.

1.5 Herramientas y tecnologías para el desarrollo del componente

Tomando como premisa que el componente para la visualización de flujos de videos será desarrollado con el objetivo de integrarse al módulo HMI del SCADA Guardián del Alba (GALBA), se utilizarán las herramientas y tecnologías empleadas en el desarrollo de dicho módulo. A continuación se caracterizan dichas herramientas y se hace una selección de las características que influyen directamente sobre el desarrollo del componente.

1.5.1 Metodología de Desarrollo

Las metodologías de desarrollo de *software* son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de *software*. En la misma se van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla. (7)

En los últimos años se han desarrollado dos corrientes fundamentales en lo referente a los procesos de desarrollo, los métodos pesados y los métodos ligeros (también llamados métodos ágiles). La diferencia fundamental entre ambos métodos consiste en que los métodos pesados pretenden lograr sus objetivos mediante el orden y la documentación, mientras que los métodos ligeros intentan mejorar la calidad del *software* mediante la comunicación directa e inmediata entre las personas que intervienen en el proceso.

Proceso Unificado de Desarrollo (RUP)

Es una metodología de desarrollo de *software* que está basada en componentes e interfaces bien definidas, y junto con el UML (Lenguaje Unificado de Modelado), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Es un proceso que puede especializarse para una gran variedad de sistemas de *software*, en diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. (8)

En su modelación define como sus principales elementos:

- Trabajadores (“quién”): Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.
- Actividades (“cómo”): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.
- Artefactos (“qué”): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- Flujo de actividades (“cuándo”): Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

Por sus características se clasifica como un método pesado, pues requiere de un grupo grande de programadores para su uso, además de que un cambio en las etapas de vida del sistema incrementaría notablemente el costo. RUP define un total de 4 fases. En cada fase se ejecutarán una o varias iteraciones (de tamaño variable según el proyecto) y dentro de cada una de ellas seguirá un modelo de cascada para los flujos de trabajo que requieren las nuevas actividades anteriormente citadas. Las fases son:

- Inicio (puesta en marcha).
- Elaboración (definición, análisis, diseño).
- Construcción (implementación).

- Transición (fin del proyecto y puesta en producción).

Extreme Programing (XP)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de *software*, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios (9). XP alienta a los desarrolladores a responder a los requerimientos cambiantes de los clientes, aún en fases tardías del ciclo del desarrollo (10). Intenta reducir la complejidad del *software* por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción. El ciclo de vida ideal para XP consiste en las siguientes fases:

- Exploración.
- Planificación de la Entrega (*Release*).
- Iteraciones.
- Producción.
- Mantenimiento.
- Muerte del Proyecto.

Esta metodología define, como bases del desarrollo de *software*, “Historias de Usuarios”, que escribe el cliente y describen escenarios sobre el funcionamiento del *software*, pueden describir el modelo, dominio, entre otros aspectos. A partir de las Historias de Usuarios (HU) y de la arquitectura perseguida, se crea un plan de entregables entre el equipo de desarrollo y el cliente.

XP no produce demasiados gastos sobre las actividades de desarrollo, y no impide el avance de los proyectos, reduce el costo del cambio en las etapas de vida del sistema y es muy recomendable para equipos de trabajo pequeños, de 2 a 15 personas, donde los programadores pueden ser ordinarios. (9)

1.5.2 Selección de la metodología a utilizar

Una vez concluido el estudio y análisis de las metodologías de desarrollo, se elaboró una tabla comparativa donde se evaluaron los principales aspectos a tener en cuenta para seleccionar la más

adecuada para el desarrollo del proyecto. Para la evaluación de los aspectos se establecieron valores que parten de 0 hasta 5 puntos.

	Ciclo de vida del proyecto	Documentación generada	Adaptación a cambios	Independencia con respecto al cliente	Centrado en la arquitectura	**Total**
XP	5	4	5	2	2	18
RUP	2	2	2	5	4	15

Tabla 1. Tabla comparativa para la selección de las metodologías.

A continuación se realiza una breve descripción de los aspectos evaluados en la tabla comparativa:

Ciclo de vida del proyecto: Se refiere al nivel de acomodamiento de la metodología al tiempo de vida que durara el proyecto.

Documentación generada: Adaptación de la documentación generada con respecto a las necesidades reales del proyecto.

Adaptación a cambios: Capacidad con la que cuenta la metodología para adaptarse a los cambios inesperados dentro del ciclo de desarrollo del proyecto.

Independencia con respecto al cliente: Autonomía de la metodología con respecto a los requerimientos que sean exigidos por el cliente.

Centrado en la arquitectura: Característica con que cuentan determinadas metodologías para realizar su desarrollo a partir de la arquitectura definida para el proyecto.

A partir del análisis realizado anteriormente, se llegó a la conclusión de utilizar XP, a continuación se exponen las principales características por las cuáles se seleccionó como metodología de desarrollo para guiar el desarrollo del componente.

- ✓ Durante su ciclo de vida genera poca documentación, evitando ser una metodología burocrática y enfocándose más en las necesidades del cliente.
- ✓ Es ideal para proyectos donde el ciclo de vida es relativamente corto.
- ✓ Tiene como prioridad satisfacer las necesidades del cliente a través de entregas continuas y tempranas.
- ✓ Los cambios a los requerimientos son bienvenidos, aún en fases tardías del desarrollo.

- ✓ Se centra fundamentalmente en entregar un producto de *software* en el menor tiempo posible y con la calidad requerida.

1.5.3 Sistema Operativo

El conjunto de programas informáticos que permite la administración eficaz de los recursos de una computadora es conocido como sistema operativo o *software* de sistema. Estos programas comienzan a trabajar apenas se enciende el equipo, ya que gestionan el *hardware* desde los niveles más básicos y permiten además la interacción con el usuario. (11)

El sistema operativo a emplear en este caso será una distribución del sistema operativo **GNU/Linux.** Una distribución es una variante de dicho sistema operativo, donde se incorporan determinados paquetes de *software* para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones hogareñas, empresariales y para servidores. Pueden ser exclusivamente de *software* libre, o también incorporar aplicaciones o controladores propietarios. Una gran parte de las herramientas básicas que completan el sistema operativo, vienen del proyecto GNU (acrónimo que significa GNU No es Unix[8]); de ahí el nombre: GNU/Linux. (12)

La distribución seleccionada para el desarrollo del componente será Debian en su versión 7.0 o *Wheezy*, la cual, se considera una de las más estables y fiables dentro del universo Linux, además de ser en esta la plataforma donde se desarrolla la nueva versión del SCADA GALBA.

1.5.4 Lenguaje de programación

Se define como lenguaje de programación al elemento dentro de la Informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que se pone a disposición del programador para que este pueda comunicarse con los dispositivos *hardware* y *software* existentes. (13)

C++

C++, es un lenguaje orientado a objetos, desarrollado en la década del 80, es un lenguaje versátil, potente y general, con buena aceptación por parte de muchos programadores profesionales, soporta directamente conceptos de la Orientación a Objetos. Desde el punto de vista de la programación orientada a objetos ofrece una buena cantidad de recursos, como son las formas de encapsulamiento, la sobrecarga de operadores y funciones, la herencia y el polimorfismo. Es tomado en cuenta para la resolución de la

[8] Sistema operativo aparecido en la década de los 60.

problemática planteada por las siguientes características: (14)

- ✓ Posee una alta difusión al ser uno de los lenguajes más empleados en la actualidad, cuenta con un gran número de usuarios y existe una amplia documentación, que servirá para un mejor entendimiento en caso de presentarse algún inconveniente durante el desarrollo del componente.
- ✓ Posee una gran versatilidad al ser un lenguaje de propósito general, por lo que se puede emplear para resolver casi cualquier tipo de problema.
- ✓ Se pueden realizar aplicaciones distribuibles sin necesidad de pagar una licencia, lo cual posibilita la realización de la aplicación en su totalidad sin tener que pagar por la utilización de este lenguaje.
- ✓ Presenta la facilidad de que es portable, propiciando que el código del componente para la visualización de flujos de videos pueda ser compilado en cualquier sistema operativo, siempre y cuando se hayan creado las bases para ello.

1.5.5 Entorno de Desarrollo Integrados (IDE)

Un entorno integrado de desarrollo (en inglés *Integrated Development Environment* o IDE) es un programa compuesto por una serie de herramientas que utilizan los programadores para escribir códigos. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos. Las herramientas que normalmente componen un entorno integrado de desarrollo son las siguientes: un editor de texto, un compilador, un intérprete, herramientas para la automatización, un depurador, un sistema de ayuda para la construcción de interfaces gráficas de usuario, y opcionalmente, un sistema de control de versiones. (15)

QtCreator 2.5.0

Como entorno de desarrollo se escoge *QtCreator*, IDE multiplataforma adaptado a las necesidades de los programadores, permite crear aplicaciones de escritorio y plataformas de dispositivos móviles, además de estar distribuido bajo los términos de la Licencia Publica General Reducida de GNU (GNU LGPL). A continuación se exponen las características más relevantes tomadas en cuenta para el desarrollo del componente.

- ✓ Combina edición, depuración, gestión de proyectos, localización y herramientas de compilación.
- ✓ Está diseñado para hacer que el desarrollo en C++ de la aplicación Qt sea más rápido y fácil; posee un avanzado editor de código C++.

- ✓ El depurador visual para C++ es consciente de la estructura de muchas clases de Qt, lo que aumenta la capacidad de mostrar los datos con claridad.
- ✓ Constituye un entorno integrado para la creación y diseño de interfaces gráficas de usuario (GUI por sus siglas en inglés) para proyectos C++.
- ✓ Los formularios son totalmente funcionales y pueden ser previamente visualizados para asegurarse de que se verá y sentirá exactamente como lo pensó el usuario. (16)

1.5.6 Marco de Trabajo

En el desarrollo de *software*, un marco de trabajo o *framework* es una estructura de soporte definida en la cual otro proyecto de *software* puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros *software* para ayudar a desarrollar y unir los diferentes componentes de un proyecto. (17)

Qt 4.8

Para el desarrollo de la presente investigación se tendrá en cuenta el *framework* Qt en su versión 4.8 ya que es un marco de trabajo multiplataforma para el desarrollo de aplicaciones con interfaz gráfica de usuario o de consola. Cuenta con abundante documentación, posee una arquitectura que soporta el uso de *plugins*[9] y hasta el momento ha sido liberado bajo dos licencias, la LGPL y la comercial, además presenta un alto rendimiento en cualquier plataforma de trabajo. Presenta varios componentes que facilitan el trabajo a los desarrolladores, los cuales se mencionan a continuación:

- ✓ Las bibliotecas Qt: clases escritas en C++ que facilitan el desarrollo de casi cualquier aplicación de *software*.
- ✓ *QtDesigner*: potente herramienta para el desarrollo de interfaces visuales.
- ✓ *QtAssistant*: acceso rápido a la documentación en caso de presentar alguna duda durante el desarrollo del componente.
- ✓ *qmake*: simplifica el proceso de construcción de proyectos en las diferentes plataformas soportadas. (18)

[9] Complemento, es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

1.5.7 Herramienta de Modelado

El lenguaje de modelado de objetos es un conjunto estandarizado de símbolos utilizados para modelar un diseño de *software* orientado a objetos. Con la utilización del lenguaje de modelado se puede llegar a obtener diseños sólidos que sirvan como guía a seguir por los analistas, desarrolladores y clientes. Logrando unificar la visión y proyección del equipo de desarrollo hacia un producto de calidad. (19)

***Visual Paradigm* 8.0**

Para el modelado de diagramas UML se empleará la herramienta Visual Paradigm, herramienta CASE (*Computer Aided Software Engineering*, Ingeniería de *Software* Asistida por Computadora) que utiliza UML como lenguaje de modelado. Está diseñada para una amplia gama de usuarios interesados en construir sistemas de *software* fiables, incluyendo actividades como ingeniería de *software*, análisis de sistemas y análisis de negocios (20).

Para la selección de esta herramienta se tomaron en cuenta las siguientes características:

- ✓ Es una herramienta que emplea las últimas notaciones de UML, ingeniería inversa, generación del código y exportación/importación XML[10].
- ✓ Está disponible en varios idiomas, licencia gratuita y comercial, es fácil de instalar y actualizar. Admite compatibilidad con las demás versiones.
- ✓ Disponibilidad en múltiples plataformas, lo cual facilita que la realización de diagramas de modelado no dependan del sistema operativo que se tenga instalado en el equipo de trabajo.
- ✓ Genera la documentación del sistema en formato PDF, HTML y DOC.

1.5.8 Biblioteca para el trabajo con flujos de video

Una biblioteca es una colección o conjunto de subprogramas usados para desarrollar *software*. En general, las bibliotecas no son ejecutables, pero sí pueden ser usadas por ejecutables que las necesitan para poder funcionar correctamente. La mayoría de los sistemas operativos proveen bibliotecas que implementan la mayoría de los servicios del sistema. Dichas bibliotecas contienen comodidades que las aplicaciones modernas esperan que un sistema operativo provea. (21)

[10] Siglas en inglés de eXtensible Markup Language, lenguaje extensible de etiquetas que permite definir lenguajes de acuerdo a las necesidades.

OpenCV

OpenCV (*Open Source Computer Vision Library*) es una biblioteca de funciones en C y C++ desarrollado por Intel[11] que proporciona un alto nivel de funciones para el procesado de imágenes, visión artificial, captura de video y visualización de imágenes. Es de código abierto, gratuita, multiplataforma (disponible para entornos MS Windows, Mac OS y Linux), está desarrollada bajo la licencia BSD (Distribución de *Software Berkeley*), rápida, de fácil uso y en continuo desarrollo. Esta biblioteca permite a los programadores crear aplicaciones poderosas en el dominio de la visión digital.

Debido a que la misma está enfocada principalmente a realizar el tratamiento de imágenes, las herramientas con los que cuenta para el manejo de flujos de video son muy ineficientes, realizando las capturas de dichos flujos, fotograma a fotograma, provocando gran consumo de recursos por parte del sistema, siendo más recomendado para otras funciones que exijan el procesado de imágenes digitales.

LibVLC

La biblioteca *libVLC* es una API[12] libre multiplataforma desarrollada por el proyecto *VideoLan*, constituye el motor principal y el proveedor de interfaz para el entorno multimedia del *VLC media player*. Esta biblioteca permite crear una amplia gama de aplicaciones utilizando las características de VLC, debido a que la misma se encuentra distribuida como biblioteca compartida. Es posible añadirla a cualquier aplicación que desee contar con la capacidad de transmitir datos *streaming*[13] a través de redes y convertir archivos multimedia en formatos distintos al original, es capaz de difundir y recibir flujos de video e información usando los protocolos, RTP *unicast* o *multicast*, RTSP, HTTP, entre otros, además de otras funcionalidades relacionadas con el trabajo con multimedia y entornos de red.

1.5.9 Selección de la biblioteca para el trabajo con flujos de video

Para determinar la biblioteca necesaria para realizar el trabajo con los flujos de video se realizó un análisis apoyado en la tabla comparativa, ver tabla número 2. En la misma se tomaron en cuenta los aspectos que más influyen en el desarrollo del componente, y fueron evaluados en una escala del 0 al 5.

[11] Compañía estadounidense líder en el sector de los microprocesadores.
[12] Interfaz de Programación de Aplicaciones por sus siglas en inglés, es el conjunto de funciones y métodos que ofrecen determinadas bibliotecas para ser utilizados por otro *software* como capa de abstracción.
[13] *Streaming* (corriente continua): Distribución de contenido multimedia a través de la red, de manera que puede ser visualizado o escuchado mientras es descargado.

	Adaptación al lenguaje escogido	Consumo de recursos de *hardware*	Trabajo con *streamings*	Procesamiento de videos	Documentación	**Total**
OpenCV	5	1	1	4	5	14
LibVLC	5	5	5	2	2	18

Tabla 2. Tabla comparativa para seleccionar las bibliotecas a utilizar para el manejo de flujos de video.

Para un mejor entendimiento de la tabla comparativa, a continuación se hace una breve descripción de los diferentes aspectos evaluados:

Adaptación al lenguaje escogido: Capacidad con que cuentan las librerías para adecuarse al trabajo con el lenguaje C++.

Consumo de recursos de *hardware*: Eficiencia en cuento al consumo de recursos de *hardware* por parte de las bibliotecas.

Trabajo con *streamings*: Facilidad con que cuentan las bibliotecas para la captura y recepción de *streamings* de video.

Procesamiento de videos: Posibilidades con las que cuentan las bibliotecas para el procesamiento de los flujos de videos.

Documentación: Información disponible para documentarse en cuanto al aprendizaje y uso de las bibliotecas.

Atendiendo al anterior análisis realizado, se selecciona la biblioteca *LibVLC* para el desarrollo del componente, además de tomarse en cuenta otras características detalladas a continuación:

- ✓ Cuenta con un *wrapper*[14] destinado para facilitar su uso e integración con el lenguaje C++ y el *framework* Qt.
- ✓ Dispone de múltiples facilidades para la gestión de flujos de video emitidos a través de la red.

[14] "Envoltorio", en la informática se refiere a una estructura de datos o *software*, la cuales pueden ser consideradas como una adaptación de ciertas sentencias de código para determinados lenguajes o aplicaciones.

- ✓ Es muy ágil a la hora de realizar la captura de flujos de videos disminuyendo dentro de lo posible los requerimientos de *hardware*.
- ✓ Permite el uso de los protocolos más comunes para la emisión y captura de video, por ejemplo, HTTP, FTP, MMS, RTSP.

1.6 Conclusiones parciales:

Una vez definido el diseño teórico de la investigación, así como todos los aspectos a tener en cuenta en este primer capítulo, se puede arribar a las siguientes conclusiones:

- ➢ El estudio realizado a las soluciones homólogas existentes, Vijeo Citect, Ingesys IT y Xilema Suria, arrojó que no sería factible utilizarlas para dar solución al problema planteado por ser aplicaciones de código cerrado, por las que hay que pagar costosas licencias, están integradas a productos de *software* específicos, funcionan con cámaras de determinados fabricantes, además de que han sido desarrolladas para plataforma Windows.
- ➢ Se identificaron las primeras técnicas empleadas para supervisar procesos a través de cámaras de video, haciendo uso de una infraestructura de red y cuáles son las principales técnicas empleadas hoy en día.
- ➢ Para el desarrollo del componente se decidió utilizar la metodología de desarrollo Extreme Programing, el lenguaje de programación C++, el marco de trabajo Qt, el entorno de desarrollo *QtCreator*, el lenguaje de modelado UML y la herramienta Visual Paradigm para el modelado del sistema.
- ➢ Para la manipulación de videos se utilizará la biblioteca *libVLC*, la misma brinda funcionalidades idóneas para la emisión y captura de video a través de la red, además de presentar un conjunto de características que la hacen idónea para dar solución a la problemática planteada.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA.

2.1 Introducción

En el siguiente capítulo se realiza la descripción del análisis y diseño del componente para la visualización de flujos de videos, se describe la propuesta para dar solución al sistema que se va a desarrollar, siguiendo la metodología de desarrollo XP como rectora del proceso de desarrollo. Se exponen las Historias de Usuarios y las Tarjetas CRC (Contenido Responsabilidad y Colaboración), además de representar las mismas en diagramas de clases, para lograr un mejor entendimiento de las relaciones entre las clases. Finalmente se define la arquitectura y los patrones de diseño que regirán el desarrollo del componente.

2.2 Propuesta de solución

Como propuesta para darle solución a la problemática planteada, se pretende desarrollar un componente que permita visualizar en tiempo real el flujo de video de una cámara IP en el Módulo HMI del SCADA GALBA, dándole al operador del SCADA la posibilidad de supervisar en tiempo real lo que ocurre en las instalaciones petroleras.

El componente deberá estar integrado en la paleta de componentes del módulo HMI, lo que brindará la opción de colocar tantos objetos como desee el encargado de realizar la configuración de los despliegues, estos componentes deben tener la capacidad de ser configurados desde el editor del módulo HMI, los parámetros más importantes a tener en cuenta a la hora de configurar el componente son: dirección IP, la cual especifica la dirección IP de la cámara que se desea visualizar; puerto, representa el puerto de escucha de la cámara; protocolo, protocolo de red por el que se emitirá el flujo de video a visualizar y finalmente la URL[15] de transmisión, que no es más que el nombre asignado al flujo de video. Además el componente debe brindar la opción de detener o reproducir el flujo de video, así como guardar fragmentos de los *streamings* en caso de que así lo decida el supervisor del módulo HMI.

Debido a que no existe la posibilidad de tener una cámara IP a tiempo completo para realizar las pruebas durante el desarrollo del componente, se hace necesario el desarrollo de una herramienta que permita la simulación de una cámara IP. La misma deberá ser capaz de simular el funcionamiento básico de una cámara IP, permitiendo configurar los parámetros necesarios para enviar un video por la red, dentro de los parámetros básicos a configurar se encuentran: el formato, la dirección IP, el puerto y el protocolo de red

[15] Localizador de recursos uniformes por sus siglas en inglés, estos representan la dirección de determinados recursos en Internet, facilitando que los navegadores encuentren y muestren de manera adecuada los mismos.

por el cual se enviará el video simulado.

2.3 Fase de exploración

El ciclo de desarrollo del *software*, según la metodología XP, comienza con la fase de exploración, en la cual, los clientes plantean a grandes rasgos las Historias de Usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán durante el desarrollo.

2.3.1 Historias de usuarios:

Las Historias de usuarios (HU), son descripciones cortas y escritas en el lenguaje del usuario, donde el nivel de detalle debe ser el mínimo posible, para que de manera sencilla se determine cuánto costará la implementación del sistema. El cliente redacta, según su consideración, los requisitos que debe tener la aplicación mediante las HU, expresando de esta manera su punto de vista en cuanto a las necesidades del sistema. A continuación se muestran las Historias de Usuarios definidas:

Historia de Usuario	
Número: 1	**Usuario:** Aplicación
Nombre historia: Capturar flujos de video.	
Prioridad en el negocio: Alta	**Nivel de complejidad:** Alta
Tiempo de Estimación: 3 semanas	**Iteración asignada:** 2
Descripción: El componente integrado en el HMI será capaz de capturar flujos de video a través de la red.	

Tabla 3. HU Capturar flujos de video.

Historia de Usuario	
Número: 2	**Usuario:** Cliente
Nombre historia: Gestionar componente cámara.	
Prioridad en el negocio: Alta	**Nivel de complejidad:** Alta
Tiempo de Estimación: 1 semana	**Iteración asignada:**1
Descripción: Se podrá colocar un componente cámara en la paleta de componentes del HMI, el cual será posible modificar o eliminar según sea el caso, además de ser posible adicionar tantos como sean necesarios al despliegue del HMI.	

Tabla 4. HU Gestionar componente cámara.

Historia de Usuario	
Número: 3	**Usuario:** Cliente
Nombre de la historia: Configurar el componente cámara.	
Prioridad en el negocio: Media	**Nivel de complejidad:** Alta
Tiempo de Estimación: 3 semanas	**Iteración asignada:** 1
Descripción: Se debe contar con la capacidad de configurar el componente cámara con la dirección IP, puerto, protocolo y URL de transmisión de donde se proveerá el flujo de video.	

Tabla 5. HU Configurar el componente cámara.

Historia de Usuario	
Número: 4	**Usuario**: Aplicación
Nombre de la historia: Visualizar el flujo de video de una cámara IP.	
Prioridad en el negocio: Alta	**Nivel de complejidad:** Alta
Tiempo de Estimación: 3 semanas	**Iteración asignada:** 2
Descripción: Un componente o varios colocados en el despliegue deben mostrar un video de cámara IP en tiempo real.	

Tabla 6. HU Visualizar flujos de video de una cámara IP.

Historia de Usuario	
Número:5	**Usuario:** Aplicación
Nombre historia: Incorporar controles de video en el componente cámara.	
Prioridad en el negocio: Media	**Nivel de complejidad:** Media
Tiempo de Estimación: 1 semanas	**Iteración asignada:** 3
Descripción: El sistema deberá incorporar controles básicos para manipular los videos que serán visualizados en el componente.	

Tabla 7. HU Incorporar controles de video en el componente cámara

Historia de Usuario	
Número:6	**Usuario:** Aplicación
Nombre historia: Exportar flujo de video	
Prioridad en el negocio: Media	**Nivel de complejidad:** Alta
Tiempo de Estimación: 2 semanas	**Iteración asignada:** 3
Descripción: El sistema deberá contar con la capacidad de guardar los videos capturados en una dirección específica en caso de que sea requerido.	

Tabla 8. HU Exportar flujo de video.

2.3.2 Requisitos no Funcionales

Un requisito no funcional es un requisito de *software* que describe no lo que el *software* hará, sino cómo lo hará. Los requisitos no funcionales son difíciles de verificar o testear, y por ello son evaluados subjetivamente (22). Se puede decir que los requisitos no funcionales constituyen la forma en la que debe actuar el sistema para que funcione de manera eficaz, atendiendo aspectos tales como la disponibilidad, flexibilidad, seguridad y facilidad de uso.

Usabilidad:

- RNF 1: Las propiedades del componente cámara podrán ser modificadas en el inspector de propiedades del HMI.
- RNF 2: El componente cámara debe disponer de algunos controles básicos para la visualización del flujo de video.

Requisito de *Hardware*:

- RNF 3: Teniendo en cuenta que el componente ha sido desarrollado para el módulo HMI del SCADA , se determinan como requisitos mínimos para el correcto funcionamiento del mismo:
    - Al menos 80 GB SATA 3.0 Gb/s 7200 RPM.

- 2 Tarjetas de Red *PCI-EXPRESS Ethernet Gigabit* de 10/100/1000 Mbps.
- Memoria RAM de 4 GB, DDR3, 1333 MHz, ECC.
- Procesador Intel Core I3 multinúcleo con velocidad igual o superior a 2.0 GHz
- Tarjeta de vídeo (profundidad mínima de color de 32 bits y resolución de 1024x768), compatible con GNU Linux Debian 6.0.6 Squeeze para posibilitar la visualización simultanea de varios despliegues en dos monitores.

2.4 Arquitectura del sistema

La arquitectura de *software* juega un papel fundamental en el diseño y la implementación de estructuras de *software* de alto nivel. Es el resultado de unir varios elementos arquitectónicos en función de satisfacer las funcionalidades y requerimientos de un sistema. Los patrones arquitectónicos constituyen un ejemplo de los elementos antes mencionados, proporcionando plantillas de trabajo que rigen el diseño de la estructura general de un sistema (8), permitiendo además que los miembros del equipo de desarrollo lo observen de una manera similar.

Para el desarrollo del componente para la visualización de flujos de video en tiempo real, se propone la utilización de la arquitectura N Capas, ya que soporta un diseño basado en niveles de abstracción crecientes, lo que a la vez permite la partición de un problema complejo en una secuencia de pasos incrementales. Admite muy naturalmente optimizaciones y refinamientos, y proporciona una amplia reutilización. Se pueden utilizar diferentes implementaciones o versiones de una misma capa en la medida que soporten las mismas interfaces de cara a las capas. (23)

2.4.1 Variante en dos capas

En el desarrollo del componente para visualizar los flujos de videos de una cámara IP en el Módulo HMI del SCADA GALBA, se definió una arquitectura en dos capas como una variante del patrón N-capas: la capa de presentación es la encargada de presentar al usuario la interfaz del componente, le permite la comunicación con el mismo, captura toda la información provista por el usuario, realizando un filtrado previo para comprobar la inexistencia de errores. Se caracteriza además, por tener una interfaz amigable, entendible y fácil de utilizar por el usuario, es la encargada de comunicarse directamente con la capa de negocio. Esta capa estará representada por la clase *UI_VideoWidget*, la misma se encarga de presentar al usuario un formulario que contiene el área donde se visualizará el video, también mostrará los controles

básicos necesarios para pausar, detener y guardar el flujo de video en caso de así lo estime conveniente el supervisor.

La capa de negocio es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso, es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa estará representada por las clases *VideoCameraIp* y *VideoWidget1*, las mismas se encargarán de realizar la captura del flujo de video desde la dirección especificada, enviarán el flujo a la capa de presentación y contendrán las funcionalidades que responden a los controles de la interfaz. A continuación se muestra una representación gráfica de la arquitectura en dos capaz.

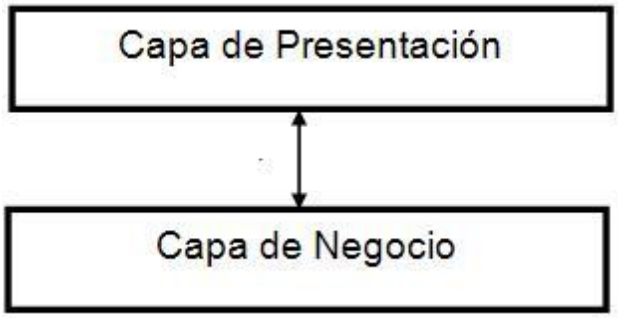

Ilustración 1. Representación gráfica de la variante en dos capas

Se escogió esta variante de la arquitectura N-Capas para el desarrollo del componente, ya que con la puesta en práctica se reducen las dependencias existentes entre las clases, de esta forma las capas situadas en los niveles más bajos no tienen conocimiento de lo que ocurre en las capas superiores, permitiendo la modificación de una capa sin afectar a las otras. La comunicación sería solo entre capas adyacentes, posibilitando que el componente tenga una arquitectura escalable, al concentrar en cada capa las funcionalidades pertinentes. Esta organización permitirá una mejor organización durante el desarrollo así como brindar un mejor soporte al sistema.

2.5 Patrones de diseño

Los patrones de diseño son la estructura para la solución a problemas comunes en el desarrollo de *software*, es decir, los mismos brindan una solución ya probada y documentada a problemas de desarrollo de *software* que están sujetos a contextos similares. Debemos tener presente los siguientes elementos de un patrón: su nombre, el problema (cuándo aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios). (24)

2.5.1 Patrones GRASP

Patrones Generales de *Software* para la Asignación de Responsabilidades (GRASP), por sus siglas en inglés, describe los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. A continuación se describen aquellos patrones utilizados de acuerdo a las soluciones que brindan dentro del sistema. (25)

- **Creador:** La creación de un objeto debe ser responsabilidad de una clase que está estrechamente relacionado con el objeto creado. Este patrón se refleja en la clase *VideoCamaraIP,* la cual es la encargada de crear las instancias necesarias para el manejo del flujo de video.
- **Controlador:** La responsabilidad para el manejo del flujo de eventos del sistema está asignada a una clase la cual puede representar todo el sistema, un dispositivo, subsistema, o un caso de uso en el que se produce el evento del sistema. En este caso el patrón se refleja en la clase *VideoCamaraIP*, la cual es la encargada de controlar los eventos del sistema.

2.5.2 Patrones GoF

Observador: Define una dependencia de uno a muchos entre objetos de forma que, cuando un objeto cambia de estado, se notifica a los objetos dependientes para que se actualicen automáticamente. (20) Al utilizar el mecanismo de señales y slots, presentes en el *framework* de desarrollo escogido, se pone de manifiesto una implementación del patrón observador.

2.6 Diseño de la solución propuesta

Uno de los elementos más importantes de la filosofía que enarbola la metodología XP, es la simplicidad en todos los aspectos, por lo cual, de acuerdo a dicho esquema, no se requiere la descripción de sistemas mediante diagramas de clases utilizando notación UML, aunque no implica que en algún momento determinado se usen. Las Tarjetas CRC (Contenido, Responsabilidad y Colaboración) son una de las técnicas empleadas para la descripción del sistema.

2.6.1 Tarjetas CRC

La principal funcionalidad de las Tarjetas CRC, es ayudar a dejar el pensamiento procedimental para incorporarse al enfoque orientado a objetos. Cada tarjeta representa una clase con su nombre en la parte superior, en la sección inferior izquierda están descritas las responsabilidades y a la derecha las clases que le sirven de soporte.

A continuación se muestra la plantilla para tarjetas CRC propuesta por la metodología XP:

Clase: Nombre de la clase que se está modelando.	
Súper Clase: Nombre de la clase padre de la herencia.	
Sub Clase(s): Nombre de las clase(s) hijas de la herencia.	
Responsabilidades: Es una descripción de alto nivel del propósito de la clase.	**Colaboraciones:** Indica las clases con las que se relaciona para cumplir su responsabilidad.

Tabla 9. Plantilla de la tarjeta CRC.

Seguidamente se muestran las principales tarjetas CRC propuestas para el desarrollo del componente.

Clase: VisorCamaraIP	
Súper Clase: Widget	
Sub Clase(s):	
Responsabilidades: Es la clase encargada de visualizar los elementos relacionados con el componente cámara.	**Colaboraciones:** *LibVLC*

Tabla 10. Tarjeta CRC *VisorCamaraIP*.

Clase: *VideoWidget*	
Súper Clase: QWidget	
Sub Clase(s):	
Responsabilidades: Es la clase encargada de capturar los flujos de video.	**Colaboraciones:** Clase *VisorCamaraIP*

Tabla 11. Tarjeta CRC *VideoWidget*.

2.6.2 Diagrama de paquetes

Por sus características la metodología XP no requiere del trabajo con los diagramas de paquetes así como otro tipo de artefactos que por lo general se generan en otras metodologías. En este caso el trabajo se apoya en el uso de un diagrama de paquetes para detallar el funcionamiento e integración del componente de video vigilancia con el HMI del SCADA GALBA. Los diagramas de este tipo muestran

como el sistema está dividido en agrupaciones lógicas, mostrando las dependencias entre las mismas. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema. Los paquetes están normalmente organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los paquetes. Cada paquete puede asignarse a un individuo o a un equipo, y las dependencias entre ellos pueden indicar el orden de desarrollo requerido. (8)

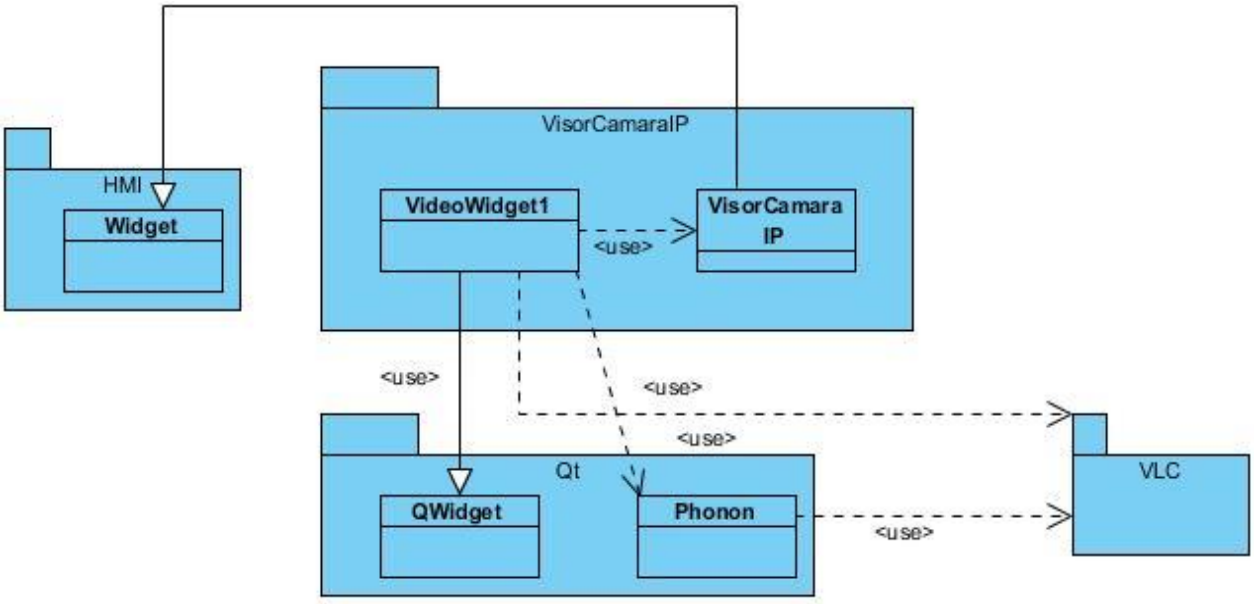

Ilustración 2. Diagrama de Paquetes.

Descripción del diagrama de paquetes

- **HMI:** Paquete que representa al módulo HMI del SCADA GALBA, contiene los componentes visuales que estarán presente en los despliegues.
- ***VisorCamaraIP:*** Paquete que representa el componente que se desea desarrollar, es el encargado de mostrar al operador los flujos adquiridos desde las cámaras IP.
- **Qt:** Paquete donde están contenidas las herramientas y utilidades que son provistas por el *framework* Qt para el desarrollo de la aplicación.
- **VLC:** En este paquete se encuentran las distintas funciones de captura y visualización de flujos de videos, las cuales son provistas por la biblioteca *libVLC*.

2.6.3 Diagrama de Clases

Los diagramas de clases permiten representar las relaciones, atributos y funcionalidades que poseen las clases del sistema. La metodología XP descarta la realización de diagramas de clase para realizar el diseño del sistema, sin embargo se decidió utilizarlas para lograr una mayor comprensión de la solución propuesta.

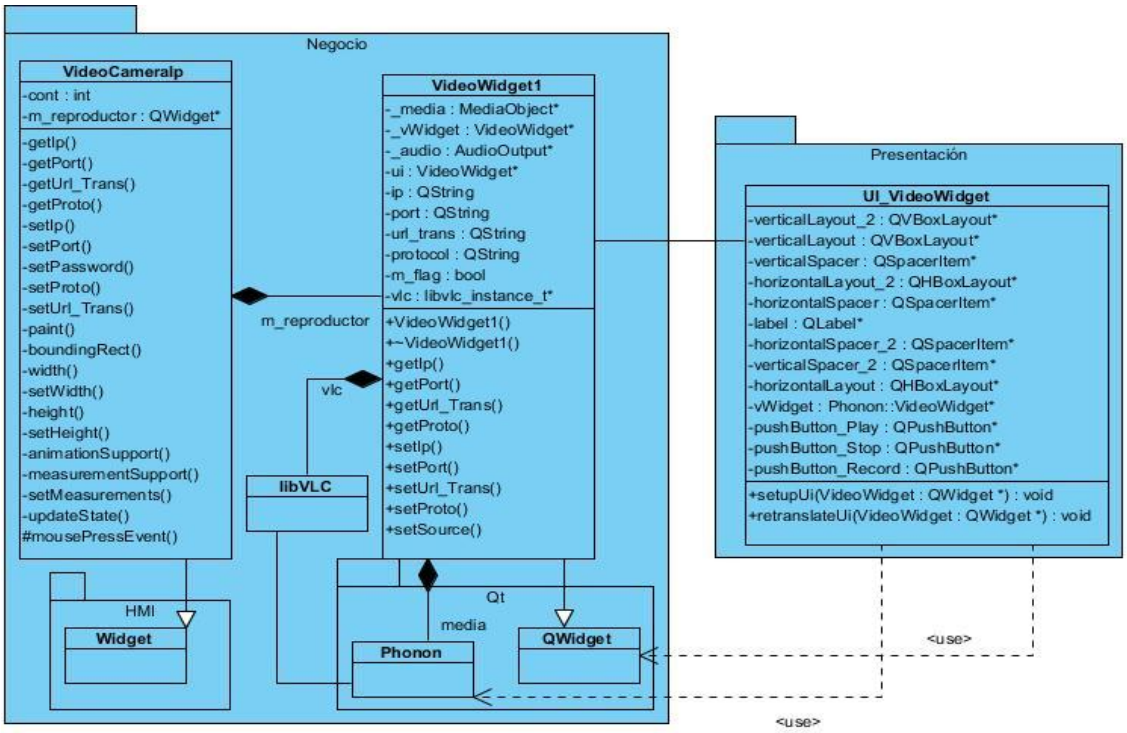

Ilustración 3. Diagrama de clases del componente para la visualización de flujos de video.

El diagrama anterior representa las principales clases arquitectónicamente significativas para darle solución al problema planteado. La clase *VideoCameraIP*, es la entidad controladora encargada de manejar los eventos propios del sistema, la clase *VideoWidget1*, es la encargada de capturar y emitir el flujo de video a la clase *Ui_VideoWidget*, la cual es la encargada de manejar los elementos visuales del componente, y presentar al usuario el video adquirido a través del módulo HMI.

2.7 Conclusiones parciales:

En este capítulo se trataron temas relacionados con el análisis y diseño del componente para la visualización de flujos de videos, utilizando la metodología de desarrollo XP como rectora del proceso de desarrollo, por lo que se puede concluir:

- La propuesta de solución permitió conocer con más detalles cómo se debería implementar el componente y cuáles eran las principales funcionalidades con las que debería cumplir.
- La descripción de las Historias de Usuarios ayudó a conocer la complejidad y prioridad de las principales funcionalidades con la que debía cumplir el componente a desarrollar.
- La selección de la arquitectura en dos capas como variante del patrón N-Capas, y la identificación de los patrones de diseño, permitió la estandarización y la contención de cambios, garantizando un mejor desempeño, robustez, portabilidad, flexibilidad y escalabilidad de la aplicación.
- El diagrama de paquetes y el diagrama de clases permitieron identificar las relaciones entre las clases del sistema, además de brindar una visión más exacta del sistema en términos de implementación.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA.

3.1 Introducción

En el presente capítulo se detalla la fase de implementación, también se ejecutan las pruebas de aceptación para validar las historias de usuario definidas anteriormente, además se expondrá el estándar de codificación utilizado para el desarrollo de la solución.

3.2 Estándar de codificación

Un estándar de codificación comprende todos los aspectos de la generación de código. Se basan en reglas o pautas con las cuales se desea obtener un código fuente que refleje un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. (26).

Para el desarrollo del componente se hace necesario usar el estándar de codificación de C++ establecido para el proyecto SCADA GALBA: (27)

- En los archivos cabecera debe incluirse el *copyright*[16] y la licencia, o una referencia a la misma, al estilo de GNU GPL.
- Se adopta el estilo de bloques de documentación de JavaDoc, el cual consiste de un bloque de comentario de estilo C.
- La descripción de los argumentos de métodos y funciones se hará en línea, es decir, luego de la declaración de cada uno de los argumentos se da una breve descripción de cada uno de ellos.
- Es importante que se especifique el nombre del autor y la fecha de creación de cualquier estructura en un código, para ello se utilizan los comandos @author y @date para el nombre del autor y la fecha respectivamente.
- El código será escrito en inglés, y la documentación en español.
- La identación por bloques será de tres (3) espacios, no se permiten tabulaciones, debido a que su representación depende de la configuración del *software* empleado para visualizar el código.
- Ninguna función debe contener más de 200 líneas de código fuente.

[16] Derechos de autor.

- Las secciones *public*, *protected* y *private* de las clases deben ser declaradas en el orden expuesto, (la sección *public* es declarada antes que la sección *protected* la cual es declarada antes de la sección *private*).

3.3 Fases de implementación

Las funcionalidades referentes al componente son desarrolladas en esta etapa, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Las HU se desglosan en tareas de programación o ingeniería, asignando a un equipo de desarrollo o programador la responsabilidad de su implementación. Teniendo en cuenta la planificación que se realizó, se llevaron a cabo 3 iteraciones, obteniéndose una solución capaz de satisfacer el problema planteado. A continuación se detallan las iteraciones.

3.3.1 Iteración I

Esta iteración tiene como objetivo dar cumplimiento a las HU 2 y 3, las cuales comprenden la gestión del componente cámara y la configuración el mismo. En la misma se sientan las bases para visualizar y definir el comportamiento del componente cámara dentro de la paleta de componentes y los despliegues del HMI.

Historia de Usuarios	Tiempo de implementación(semanas)	
	Estimación	Real
Gestionar componente cámara.	1	1
Configurar el componente cámara.	2	2

Tabla 12. HU abordadas en la Iteración I.

Tarea de Ingeniería	
No. de la Tarea:1	**No. de la HU:**2
Nombre de la Tarea: Registrar un elemento cámara en la paleta de componentes del HMI.	
Tipo de Tarea: Desarrollo	**Puntos estimados:**0.5
Fecha de inicio: 31/03/2014	**Fecha del fin:** 2/04/2014
Programador responsable: Eduardo Grabiel Luzúa Gómez.	
Descripción: Se definen las funcionalidades necesarias para que el componente cámara sea visualizado en la paleta de componentes que forma parte de los despliegues del HMI del SCADA GALBA.	

Tabla 13. Tarea No. 1 de la HU No. 2

Tarea de Ingeniería	
No. de la Tarea:2	**No. de la HU:** 2
Nombre de la Tarea: Definir el comportamiento del componente cámara dentro del Editor del HMI.	
Tipo de Tarea: Desarrollo	**Puntos estimados:**0.5
Fecha de inicio: 2/04/2014	**Fecha del fin:**4/04/2014
Programador responsable: Eduardo Grabiel Luzúa Gómez.	
Descripción: Se establecen las funcionalidades básicas para la manipulación del componente cámara, las cuales son: adicionar, borrar y editar.	

Tabla 14. Tarea No. 2 de la HU No. 2

Tarea de Ingeniería	
No. de la Tarea:3	**No. de la HU:**3
Nombre de la Tarea: Definir los eventos relacionados con la manipulación del componente cámara dentro de un despliegue en el HMI.	
Tipo de Tarea: Desarrollo	**Puntos estimados: 1**
Fecha de inicio: 7/04/2014	**Fecha del fin:**11/04/2014
Programador responsable: Eduardo Grabiel Luzúa Gómez.	
Descripción: Se desea que los eventos propios de los elementos de la paleta de componentes del HMI formen parte del componente cámara, en este caso desplazar el mismo dentro del despliegue en cualquier posición que se estime conveniente.	

Tabla 15. Tarea No. 3 de la HU No. 3

Tarea de Ingeniería	
No. de la Tarea:4	**No. de la HU:**4
Nombre de la Tarea: Definir la clase genérica que se encargará de manejar la configuración las propiedades del componente cámara.	
Tipo de Tarea: Desarrollo	**Puntos estimados:**1
Fecha de inicio: 21/04/2014	**Fecha del fin:**25/04/2014
Programador responsable: Eduardo Grabiel Luzúa Gómez.	
Descripción: Se definen las propiedades a manejar en el componente cámara a través de una clase genérica encargada de la configuración del mismo.	

Tabla 16. Tarea No.4 de la HU No. 4

3.3.2 Iteración II

En esta iteración se pretende dar cumplimiento a las historias de usuario 1 y 5, las cuales se encargarán de capturar y visualizar los flujos de video de una cámara IP. Con la realización de esta iteración, el componente creado previamente será capaz de capturar y visualizar flujos de videos desde cámaras IP en los despliegues del HMI.

Historia de Usuarios	Tiempo de implementación(semanas)	
	Estimación	Real
Capturar flujos de video	2	2
Visualizar el flujo de video de una cámara IP	1	1

Tabla 17. HU abordadas en la Iteración II.

Tarea de Ingeniería	
No. de la Tarea:5	**No. de la HU:** 1
Nombre de la Tarea: Definir la clase genérica encargada de manejar las instancias para la captura del video e integrarlo al HMI.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 0.5
Fecha de inicio: 28/04/2014	**Fecha del fin:** 30/04/2014
Programador responsable: Eduardo Grabiel Luzúa Gómez.	
Descripción: Se define la clase genérica que se ocupará de manejar las instancias necesarias para la captura de los flujos de video, así como de integrar el mismo con la interfaz del HMI.	

Tabla 18. Tarea No.5 de la HU No. 1

Tarea de Ingeniería	
No. de la Tarea:6	**No. de la HU:** 1
Nombre de la Tarea: Definir la clase encargada de realizar la captura de video.	
Tipo de Tarea: Desarrollo	**Puntos estimados:**1.5
Fecha de inicio: 1/05/2014	**Fecha del fin:**9/05/2014
Programador responsable: Eduardo Grabiel Luzúa Gómez.	
Descripción: Se define la clase que permitirá capturar el flujo de video a través de la red.	

Tabla 19. Tarea No.6 de la HU No. 1

Tarea de Ingeniería	
No. de la Tarea:7	**No. de la HU:** 5
Nombre de la Tarea: Definir la funcionalidades necesarias para permitir la visualización del flujo de video.	
Tipo de Tarea: Desarrollo	**Puntos estimados:**1
Fecha de inicio: 12/05/2014	**Fecha del fin:**16/05/2014
Programador responsable: Eduardo Grabiel Luzúa Gómez.	
Descripción: Se definen las funcionalidades necesarias para permitir que el flujo de video pueda ser visualizado en el componente colocado en el despliegue del SCADA.	

Tabla 20. Tarea No. 7 de la HU No.5

3.3.3 Iteración III

En la presente iteración se le dará solución a las historias de usuario 6 y 7, las cuales comprenden, exportar los flujos de video a una dirección específica, así como realizar los ajustes básicos en los flujos

de video. De manera general, en esta iteración se establecen los controles básicos para el manejo de los flujos de video, además se le proporcionará al componente la capacidad de guardar los flujos de videos en la dirección que el usuario estime conveniente.

Historia de Usuarios	Tiempo de implementación(semanas)	
	Estimación	Real
Incorporar controles de video en el componente cámara	1	1
Exportar flujos de video.	2	2

Tabla 21 HU abordadas en la Iteración III.

Tarea de Ingeniería	
No. de la Tarea: 8	**No. de la HU:** 6
Nombre de la Tarea: Definir las funcionalidades necesarias para incorporar controles en los flujos de video.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 19/05/2014	**Fecha del fin:** 23/05/2014
Programador responsable: Eduardo Grabiel Luzúa Gómez.	
Descripción: Se define en la clase correspondiente cuáles son las funcionalidades necesarias para realizar controles básicos en los flujos de video, en este caso debe permitirse iniciar o detener la captura de los videos en caso de que el cliente lo estime conveniente.	

Tabla 22. Tarea No. 8 de la HU No. 6.

Tarea de Ingeniería	
No. de la Tarea: 9	**No. de la HU:** 7
Nombre de la Tarea: Definir las funcionalidades necesarias para exportar los flujos de video a una dirección especifica.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 26/05/2014	**Fecha del fin:** 6/05/2014
Programador responsable: Eduardo Grabiel Luzúa Gómez.	
Descripción: Se define en la clase correspondiente cuáles son las funcionalidades necesarias para exportar los flujos de video a una dirección específica.	

Tabla 23. Tarea No. 9 de la HU No. 7.

3.4 Diagrama de componente

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en un modelo de diseño. Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de *software*, sean estos componentes de código fuente, binarios o ejecutables. Algunos estereotipos de componentes son los siguientes: (28)

- <<*executable*>> programa que puede ser ejecutado en un nodo.
- <<*file*>> fichero que contiene código fuente o datos.
- <<*library*>> biblioteca, estática o dinámica.
- <<*table*>> tabla de base de datos.

A continuación se representa el diagrama de componentes de código fuente del componente para la visualización de flujos de videos desde una cámara IP.

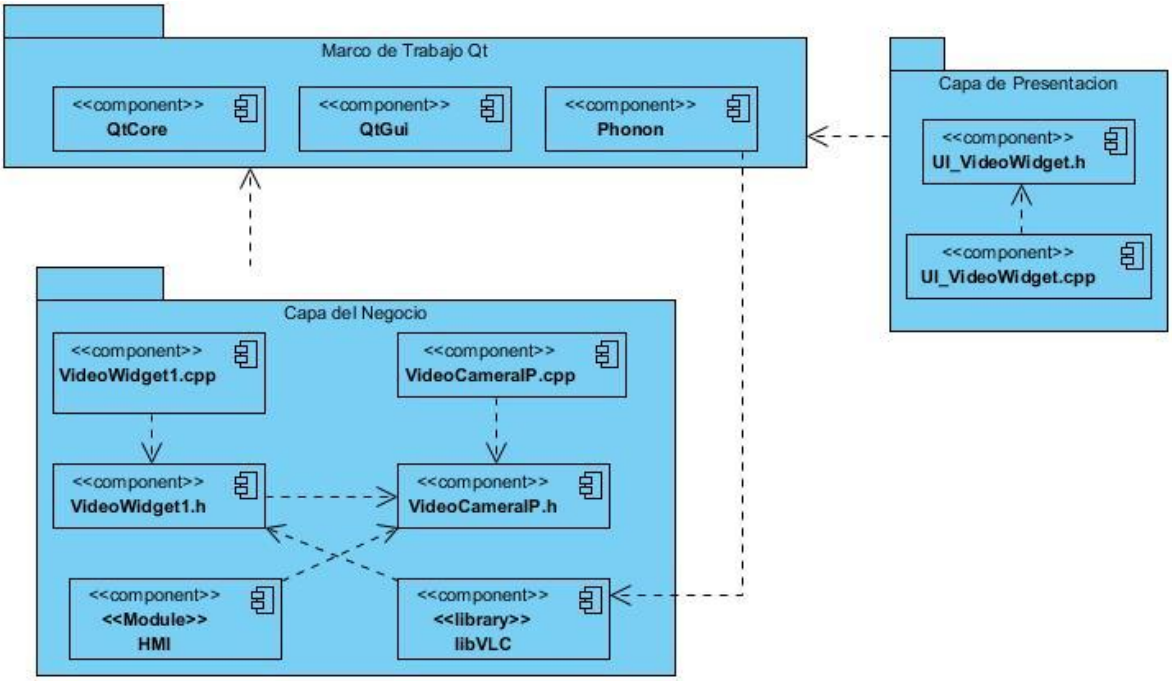

Ilustración 4. Diagrama de componentes del código fuente.

A continuación se representan las principales relaciones de dependencia entre los componentes más relevantes:

- ✓ El componente ***Phonon*** se encuentra relacionado con el componente ***libVLC***, debido a que ***Phonon*** instancia elementos de ***libVLC*** para poder realizar el trabajo con archivos multimedia.
- ✓ El componente **HMI** se relaciona con la clase controladora ***VideoCameraIP***, debido a que ***VideoCameraIP*** utiliza elementos del módulo **HMI** para representar el componente en la paleta de componentes, configurar el mismo en el inspector de propiedades y permitir que pueda ser ubicado en los despliegues.
- ✓ El componente ***VideoCameraIP*** se relaciona con la clase ***VideoWidget1***, en este caso la clase ***VideoCameraIP*** instancia la clase ***VideoWidget1***.

3.5 Despliegue del sistema

Los diagramas de despliegue muestran la disposición física de los nodos que componen el sistema y el reparto de los componentes sobre dichos nodos, modelando la topología de *hardware* sobre la cual se ejecutará el sistema. Estos nodos representan un recurso computacional que generalmente cuenta con memoria y capacidad de procesamiento. (29)

En la imagen mostrada a continuación se puede apreciar el nodo PC Supervisión, que representa el ordenador donde estará instalado el módulo HMI y por consiguiente el componente para la visualización de los flujos de video. El nodo cámara IP, representa la cámara a la que se encuentra conectado el componente para visualizar lo que ocurre en tiempo real. Esta conexión se establece mediante el protocolo de red TCP/IP.

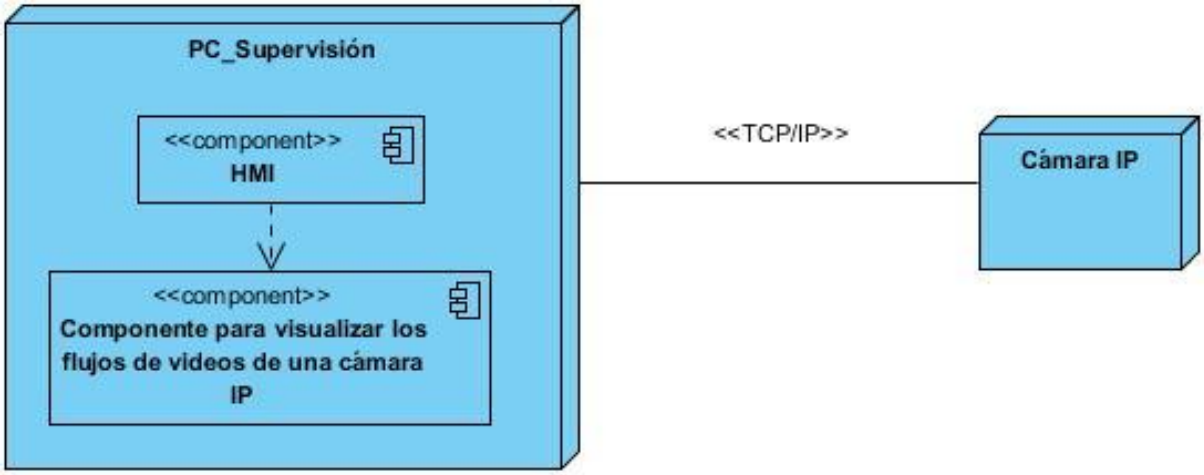

Ilustración 5. Diagrama de despliegue.

3.6 Pruebas

La fase de pruebas dentro de la Metodología XP representa uno de los momentos más importantes en el desarrollo de la misma. El uso de las pruebas permite comprobar el funcionamiento del código que se va implementando, permitiendo aumentar la calidad de los sistemas y reduciendo el número de errores que se detectan, además disminuyen el tiempo transcurrido entre la aparición de un error y su detección. Por último permite aumentar la seguridad y evita efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones.

En la metodología XP las pruebas son divididas en dos grupos: las pruebas unitarias y las pruebas de aceptación. Las pruebas unitarias son llevadas a cabo por los programadores, los cuales verifican el código automáticamente, por otro lado las pruebas de aceptación se destinan a comprobar que cada Historia de Usuario cumpla con la funcionalidad asignada y que esta satisfaga las necesidades del cliente. (30)

3.6.1 Pruebas de aceptación

Las pruebas de aceptación son consideradas más importantes con respecto a las unitarias, por el hecho de que representan la satisfacción con el producto desarrollado, el final de una iteración y el comienzo de la siguiente, siendo el cliente la persona idónea para diseñar las pruebas a ejecutar. (30)

La puesta en práctica de las pruebas de aceptación permitió que se comprobaran las funcionalidades definidas. A continuación se muestran los casos de pruebas satisfactorios diseñados durante la tercera Iteración de las pruebas, los cuales permitieron validar el correcto funcionamiento del componente.

Caso de Prueba de Aceptación	
Código: HU2_P1	**Historia de Usuario:** 2
Nombre: Capturar flujos de video a través de la red.	
Descripción: Prueba para realizar la captura de flujos de video a través de la red.	
Condiciones de Ejecución: Componente cámara correctamente configurado.	
Entradas / Pasos de Ejecución: Parámetros de configuración del componente cámara.	
Resultado Esperado: El componente captura correctamente el flujo de video.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 24. Prueba de aceptación para la HU “Capturar flujos de Video”.

Caso de Prueba de Aceptación	
Código: HU3.1_P1	**Historia de Usuario:** 3.1
Nombre: Comprobar que se adiciona el componente cámara al Despliegue del HMI SCADA.	
Descripción: Se prueba la adición del componente cámara al despliegue del HMI SCADA.	
Condiciones de Ejecución: Se necesita un despliegue en el cual adicionar el componente.	
Entradas / Pasos de Ejecución: Arrastrar el componente(s) a la zona del despliegue.	
Resultado Esperado: Se adiciona correctamente el componente al despliegue del HMI SCADA.	
Evaluación de la Prueba: Satisfactorio.	

Tabla 25. Prueba de aceptación para la HU “Gestionar componente cámara”.

Caso de Prueba de Aceptación	
Código: HU3.2_P1	**Historia de Usuario:** 3.2
Nombre: Modificar los datos del componente cámara en el despliegue del HMI SCADA.	
Descripción: Se comprueba que se modifican los datos del componente de manera correcta.	
Condiciones de Ejecución: Valor de los datos a modificar en el componente.	
Entradas / Pasos de Ejecución: Datos del componente.	
Resultado Esperado: Se modificaron correctamente los datos del componente.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 26. Prueba de aceptación para la HU “Modificar componente”.

Caso de Prueba de Aceptación	
Código: HU3.3_P1	**Historia de Usuario:** 3.3
Nombre: Eliminar el componente cámara del Despliegue del HMI SCADA.	
Descripción: Se comprueba que se elimina correctamente el componente cámara del Despliegue.	
Condiciones de Ejecución: Debe existir un componente cámara en el despliegue.	
Entradas / Pasos de Ejecución: Se selecciona el componente en el despliegue y se procede a eliminar.	
Resultado Esperado: El componente cámara se elimina correctamente.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 27. Prueba de aceptación para la HU “Eliminar componente.”

Caso de Prueba de Aceptación	
Código: HU6_P1	**Historia de Usuario:** 6
Nombre: Visualizar el flujo de video de una cámara.	
Descripción: Se prueba que se visualice el flujo de video proveniente de una cámara IP.	
Condiciones de Ejecución: La cámara debe estar encendida y conectada a la red.	
Entradas / Pasos de Ejecución: Introducir los datos necesarios para configurar el componente.	
Resultado Esperado: Se visualiza correctamente el flujo de video en el componente.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 28. Prueba de aceptación para la HU “Visualizar flujo de video de una cámara IP.”

Caso de Prueba de Aceptación	
Código: HU5_P1	**Historia de Usuario:** 5
Nombre: Configurar el componente cámara.	
Descripción: Se prueba que los parámetros introducidos en el componente cámara sean válidos.	
Condiciones de Ejecución: Debe existir un componente cámara en el despliegue y debe ser configurado.	
Entradas / Pasos de Ejecución: Se introducen los datos en el inspector de propiedades del HMI SCADA.	
Resultado Esperado: Se configura correctamente el componente en base a los datos que le fueron introducidos.	
Evaluación de la Prueba: Satisfactoria	

Tabla 29. Prueba de aceptación para la HU “Configurar el componente cámara”.

3.7 Resultado obtenido al aplicar las pruebas al *software*

Durante la fase de prueba se realizaron tres iteraciones al componente para la visualización de flujos de videos de una cámara IP, en el transcurso de las mismas se contó con la presencia del arquitecto principal del SCADA GALBA. A continuación se explican las inconformidades detectadas por parte del arquitecto durante las iteraciones de prueba:

Iteración 1:

- ✓ Una vez desarrollada la primera versión del componente, la solución propuesta no satisfacía en su totalidad las necesidades y requerimientos del arquitecto, debido a que el componente mostraba los flujos de videos en un despliegue diferente al que monitorizaba los eventos principales del SCADA. La inconformidad presentada por parte del arquitecto fue corregida, lográndose que los flujos de videos adquiridos desde las cámaras IP fueran visualizados dentro del despliegue principal junto a los restantes elementos.

Iteración 2:

- ✓ Se detectó que el componente no estaba mostrando correctamente los flujos de videos, el error se producía debido a que la cadena que proveía al componente la dirección, puerto, protocolo y nombre de los flujos de video, no se concatenaba correctamente. Además se detectó que al efectuarse cambios en las configuraciones de los componentes ubicados en el despliegue del HMI, no se actualizaban correctamente dichas configuraciones, debido a que las funcionalidades encargadas de actualizar los datos enviados desde el inspector de propiedades del módulo HMI, no se encontraban correctamente implementadas, estas inconformidades fueron corregidas, lográndose que fueran mostrados correctamente los videos dentro del componente, así como, actualizados los cambios realizados a la configuración del mismo.

Iteración 3:

- ✓ Durante la tercera iteración no se detectaron inconformidades.

La siguiente imagen muestra una representación gráfica de los resultados obtenidos durante las pruebas aplicadas al sistema:

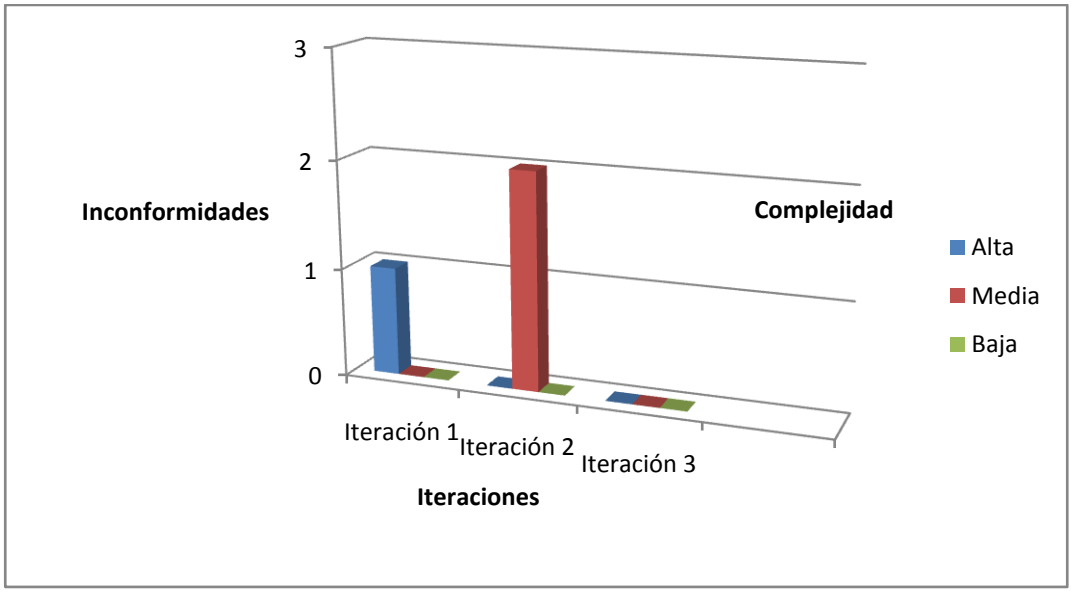

Ilustración 6. Resultado de las pruebas

Una vez solucionadas todas las inconformidades detectadas durante las tres iteraciones de las pruebas realizadas al componente para visualizar los flujos de video de una cámara IP en el módulo HMI del

SCADA GALBA, la aplicación recibió la aprobación del arquitecto principal del SCADA, así como la del jefe de la línea Seguridad. La aprobación fue emitida a través de una carta de validación, en la cual se expresa la conformidad de los involucrados en el proceso de aceptación, y el correcto funcionamiento del componente desarrollado, ya que cumple con todos los requerimientos definidos para el desarrollo del componente. Por lo tanto, el componente puede ser integrado al SCADA GALBA y desplegado como parte de las futuras versiones de dicho producto.

3.8 Resultado obtenido

Como resultado del desarrollo del componente para la visualización de videos en tiempo real dentro de los despliegues del HMI del SCADA GALBA, se obtuvo una aplicación que brinda la posibilidad de que sean observados los eventos que ocurren en las instalaciones petroleras, se representó el mismo dentro de la paleta de componentes del despliegue, permitiendo que forme parte de todo el conjunto de estereotipos que pueden ser utilizados en los diferentes despliegues, contando además con la posibilidad de que pueda ser configurado en el inspector de propiedades del editor presente en el HMI. Los parámetros a configurar son: puerto de escucha de la cámara, dirección IP, protocolo de red por el cual se transmitirá el flujo de video y nombre del flujo de video o URL de transmisión. El componente cuenta con la posibilidad de detener o reproducir el flujo de video además de guardar flujos de video en una dirección específica en caso de que así lo determine el operador.

Como valor agregado se obtuvo un simulador de cámaras IP con el cual fue posible realizar las pruebas iniciales al componente. Este simulador cuenta con la posibilidad de poder configurar los parámetros básicos que generalmente son configurados en las cámaras reales, por ejemplo es posible determinar el formato en que será codificado el flujo de video a transmitir, la dirección IP a la que se podrán conectar los clientes, así como el puerto y el protocolo de red.

3.9 Conclusiones parciales

En este capítulo se realizó la descripción de la fase de implementación y las pruebas realizadas al sistema, así como una representación gráfica de los principales artefactos generados durante la fase de implementación, arribando a las siguientes conclusiones:

- Las descripción de las tareas de ingeniería permitió guiar el proceso de desarrollo, evidenciando cuáles eran las funcionalidades a implementar y el tiempo invertido en cada una de ellas.

- El diagrama de componentes permitió comprender la interacción entre los componentes que conforman el sistema y mostrar las dependencias que existen entre los mismos.
- La fase de pruebas permitió corregir las no conformidades detectadas que afectaban el correcto funcionamiento del sistema, validando que el componente es completamente funcional y que está listo para integrarse al módulo HMI de SCADA GALBA.

CONCLUSIONES GENERALES

Con la realización de la presente investigación se le dio cumplimiento al objetivo general así como a las tareas de la investigación, arribando a las siguientes conclusiones:

- El estudio de los conceptos asociados al dominio del problema, permitió la comprensión de los principales términos que se manejaron durante la investigación.
- El estudio de las soluciones similares existentes, ayudó a determinar las principales características positivas y negativas de los sistemas actuales, para visualizar procesos industriales dentro de los sistemas SCADA a través de cámaras IP, tomando como premisas para el desarrollo del componente, las características positivas de estos sistemas.
- Para el desarrollo del componente se decidió utilizar la metodología de desarrollo *Extreme Programing*, el lenguaje de programación C++, el marco de trabajo Qt, el entorno de desarrollo *QtCreator*, el lenguaje de modelado UML y la herramienta *Visual Paradigm* para el modelado del sistema.
- La descripción y el diseño del componente para visualizar los flujos de videos de una cámara IP en el Módulo HMI del SCADA GALBA, permitió crear las bases para la implementación de dicho sistema, evaluando y restringiendo cada posibilidad de error.
- Las pruebas realizadas al componente ayudaron a corregir inconformidades que no se tuvieron en cuenta durante la implementación, permitiendo refinar el producto y liberándolo de inconformidades.
- Como resultado de la investigación se obtuvo un componente que permite supervisar en tiempo real, la quema del gas resultante de la extracción de petróleo y las distintas áreas de las plantas petroleras, el mismo está integrado al módulo HMI del SCADA GALBA, lo que ayuda a que la supervisión de los procesos industriales sea más centralizada. Finalmente es importante destacar que para el desarrollo del componente solo se utilizaron herramientas y tecnologías libres.

RECOMENDACIONES

Durante el desarrollo del presente trabajo han surgido nuevas propuestas que podrían implementarse en una versión futura, de forma tal que se lograría incrementar las funcionalidades del componente, y el sistema en general, ampliando de esa manera su funcionamiento, para lo cual se recomienda:

- Agregar funcionalidades que permitan realizar tratamiento de imágenes para mejorar los sistemas de alerta.
- Incorporar al sistema nuevas funcionalidades que permitan controlar las cámaras IP desde el módulo HMI.
- Incorporar un gestor de video que permita organizar todo el trabajo referente a los flujos de videos.

REFERENCIAS BIBLIOGRÁFICAS

1. **Definicion.de.** Definicion.de. [En línea] 2014. http://definicion.de/video/.

2. **del Bimbo, Alberto.** *Bases de Datos de Imágenes y Videos.* Ciudad de la Habana : s.n., 2005.

3. **Inteco.** *Inteco.* [En línea] 3 de Junio de 2006. http://www.inteco.es/pressRoom/Prensa/Actualidad_INTECO/guia_videovigilancia/.

4. **Real Academia Española.** *Real Academia Española.* [En línea] 2014. http://www.rae.es/.

5. **Electronic Dreams.** *Electronic Dreams.* [En línea] 2014. http://www.camara-ip.es.

6. **Durañona, Axel Rodríguez.** Gestor. Módulo para el Sistema de Video Vigilancia. La Habana : s.n., 2012.

7. **Universidad de Murcia.** Universidad de Murcia. [En línea] 2014. http://www.um.es/.

8. **Pressman, Roger S.** *Ingenieria de Software. Un enfoque Práctico.* 2010.

9. **Beck, K.** *Extreme Programming Explained. Embrace Change.* s.l. : Pearson Education, 1999.

10. **Joskowicz, José.** Reglas y Prácticas en eXtreme Programming. 2008.

11. **Definicion.de.** Definicion.de. [En línea] 2014. http://definicion.de/sistema-operativo/.

12. **Free Software Foundation, Inc.** Free Software Foundation, Inc. [En línea] 2009. http://www.gnu.org/home.es.html.

13. **definicion.org.** definicion.org. [En línea] 2014. http://www.definicion.org/lenguaje-de-programacion.

14. **Stroustrup, Bjarne.** *The C++ Programming Language (Third Edition).* s.l. : Addison-Wesley, 1997.

15. *Sitio Web de la E.U. de Ingeniería Técnica Informática de oviedo.* [En línea] 2009. http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf.

16. **Qt Project.** Qt Creator. [En línea] 2014. http://qt-project.org/wiki/Category:Tools::QtCreator_Spanish.

17. **CodeBox.** *CodeBox.* [En línea] 2011. http://www.codebox.es/. .

18. **Techeral.** *Techeral.* [En línea] 2012. http://techerald.com.

19. **Craig, Larman.** *UML y PATRONES. Introducción al análisis y diseño orientado a objeto.* 2008.

20. **Visual Paradigm.** Visual Paradigm. [En línea] 2014. http://www.visual-paradigm.com/product/vpuml/.

21. **ALEGSA.** [En línea] 2014. http://www.alegsa.com.ar/Dic/biblioteca.php.

22. **González, Oscar Díaz.** *Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software.* 2007.

23. **Kicillof, Carlos Reynoso y Nicolás.** *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft.* 2009.

24. **Microsoft Corp.** *MSDN-the microsoft developer network.* [En línea] 2014. http://msdn.microsoft.com/es-es/library/bb972240.aspx.

25. **Sánchez, Esther Guerra.** [En línea] 2009. http://arantxa.ii.uam.es/~eguerra/docencia/0809/09%20Observer.pdf.

26. **Microsoft Corp.** MSDN-the microsoft developer network. *Microsoft Corporation.* [En línea] 2014. http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx.

27. **Chávez Lorenzo, Ariel.** *Estandar de Codificación para C++. Plataforma de Supervisión y Control Guardián del Alba.* La Habana : s.n., 2012.

28. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software.* Madrid : Pearson Educación, 2000.

29. **Guerra, Javier Lorié.** *Capa de acceso a datos para dispositivos Bristol.* Ciudad de la Habana : s.n., 2011.

30. **Gutiérrez, Javier J., Escalona, M. J., Mejías, M. y Torres, J.** [En línea] 2006. http://www.lsi.us.es/~javierj/investigacion_ficheros/PSISEXTREMA.pdf).

BIBLIOGRAFÍA

1. **Guerra Sánchez, Esther.** Patrones de Diseño. *Patron de Comportamiento Observer.* Madrid : s.n., 2008.

2. **Echeverry Tobón, Luis Miguel and Elena, Delgado Carmona Luz.** Caso práctico de la metodología ágil XP al desarrollo de software. Pereira : s.n., 2007.

3. **INGETEAM.** *INGETEAM.* [Online] 2014. http://www.ingeteam.com/Portals/0/Catalogo/Producto/Documento/PRD_483_Archivo_ptd69-ficha-ingesys-it-soluciones-de-seguridad.pdf.

4. **Schneider Electric.** *Schneider Electric.* [Online] 2014. http://www.schneider-electric.com.mx/sites/mexico/es/productos-servicios/automatizacion-control/noticias/viewer-noticias.page?c_filepath=/templatedata/Content/News/data/es/shared/automation_control/general_information/2010/11/1110_04scada.xml.

5. **Petroleos de Venezuela.** *PDVSA.* [Online] 2005. http://www.pdvsa.com/PESP/Pages_pesp/aspectostecnicos/gasnatural/comoseprocesa.html.

6. **Ardines, Isabel Anayansi, Fong, Ana Teresa De and Ruíz, Eric Edgardo.** www.monografias.com. [Online] 2013. http://www.monografias.com/trabajos5/petroleo/petroleo.shtml.

7. **COMTEL.** COMTEL. [Online] 2013. http://www.comtel.com.sv/sol_video_vigilancia.html.

8. **34Telecom.** 34Telecom. [Online] 2014. http://www.34t.com/box-docs.asp?doc=551.

9. **MUCCIO, ING. CARLOS DI.** *INDIGOVISION.* [Online] Julio 14, 2011. http://www.slideshare.net/LogicalisLatam/indigovision-revejulio2011110719160512phpapp01.

10. **D-Link.** D-LINK Europe Ltd. [Online] 2012. www.dlink.com/es/es/business-solutions/ip-surveillance.

11. **Revista Negocios de Seguridad.** Revista Negocios de Seguridad. [Online] www.rnds.com.ar/articulos/031/RNDS_084W.pdf.

12. **Axis Communications AB.** Axis Communications AB. [Online] 2013. http://www.axis.com/es/products/video/about_networkvideo/evolution.htm.

13. **www.creatienda.de.** [Online] Agosto 4, 2013. http://www.creatienda.de/index.php/historia-cronologia-

internet-web-cell/historia-camara-ip-radio-movil-celular-gsm-smartphone/historia-camara-ip-en-la-web.html.

14. **Nexobit.com**. [Online] 2011. http://nexobit.com/2011/05/06/la-visita-del-creador-de-la-camara-ip-y-un-poco-de-historia/.

15. **Bradski, Gary and Kaehler, Adrian.** Learning OpenCV. Computer Vision with the OpenCV Library. O´Reilly: s.n., 2008.

ANEXOS

Anexo 1: Entrevista.

- A su consideración ¿cuáles son los motivos por los cuales es necesaria la implementación de un componente de video vigilancia?
- ¿Cuáles son los aspectos que usted considera debería tener un componente para la visualización de videos emitidos por cámaras IP?
- ¿Qué herramientas me recomienda para el desarrollo de dicho componente?
- ¿Cuáles serían los principales problemas a resolver a la hora de capturar los flujos?
- A la hora de integrar el componente cámara en la interfaz del HMI SCADA, ¿Cuáles son las condiciones básicas para permitir un manejo eficiente del mismo?

Como resultado de las entrevistas realizadas se establecieron los principales problemas a tratar y la propuestas de herramientas. Actualmente existen deficiencias en los sistemas de seguridad por cámara, entre otras, el operador encargado de supervisar la seguridad de los sistemas de SCADA no cuenta con un mecanismo lo suficientemente práctico para controlar los monitores que muestran lo que ocurre en tiempo real en las instalaciones al mismo tiempo los sistemas de alertas propios del SCADA, por lo tanto sería muy importante integrar ambos elementos facilitando el manejo de situaciones de emergencia y evitar distracciones por parte del operador.

Por otro lado a la hora de analizar las herramientas a emplear se sugirió desechar el uso de la biblioteca *OpenCV* para el trabajo con videos, gran consumidora de recursos y con muchas limitantes a la hora de manejar flujos de video, en su lugar existe otra alternativa más potente para el trabajo con *streamings* de video, la *libVLC*, dicha biblioteca es mucho más práctica, menos exigente en cuanto a requisitos de *hardware* se refiere, las misma cuenta con soporte en los repositorios de la distribución empleada en el desarrollo de los módulos del SCADA, Debian, también cuenta con un *wrapper* desarrollado específicamente para el trabajo con el *QtCreator* uno de los principales IDE de desarrollo empleado en el sistema SCADA.

Otro de los aspectos tratados se basó en sugerencias respecto a la interfaz que sería utilizada, cómo se lograría una mejor integración del componente con el módulo HMI del SCADA, las funcionalidades básicas con las que se debían cumplir para garantizar un manejo eficiente por parte de los operadores, así como

cuáles eran las condiciones mínimas para lograr la comunicación con las cámaras IP.

A grandes rasgos, en las entrevistas se permitió definir cuál sería la situación problemática a tratar y el alcance que debería el desarrollo del componente en su primera etapa, definir los rasgos principales para su integración con el módulo HMI del SCADA, sugerir cuales serían las tecnologías y herramientas a analizar para la implementación posterior.