

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 6



**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Título: Herramienta de análisis de videos grabados en el Sistema de video-
vigilancia Xilema Suria.

Autores: Andy Yonnié Martel Carpio

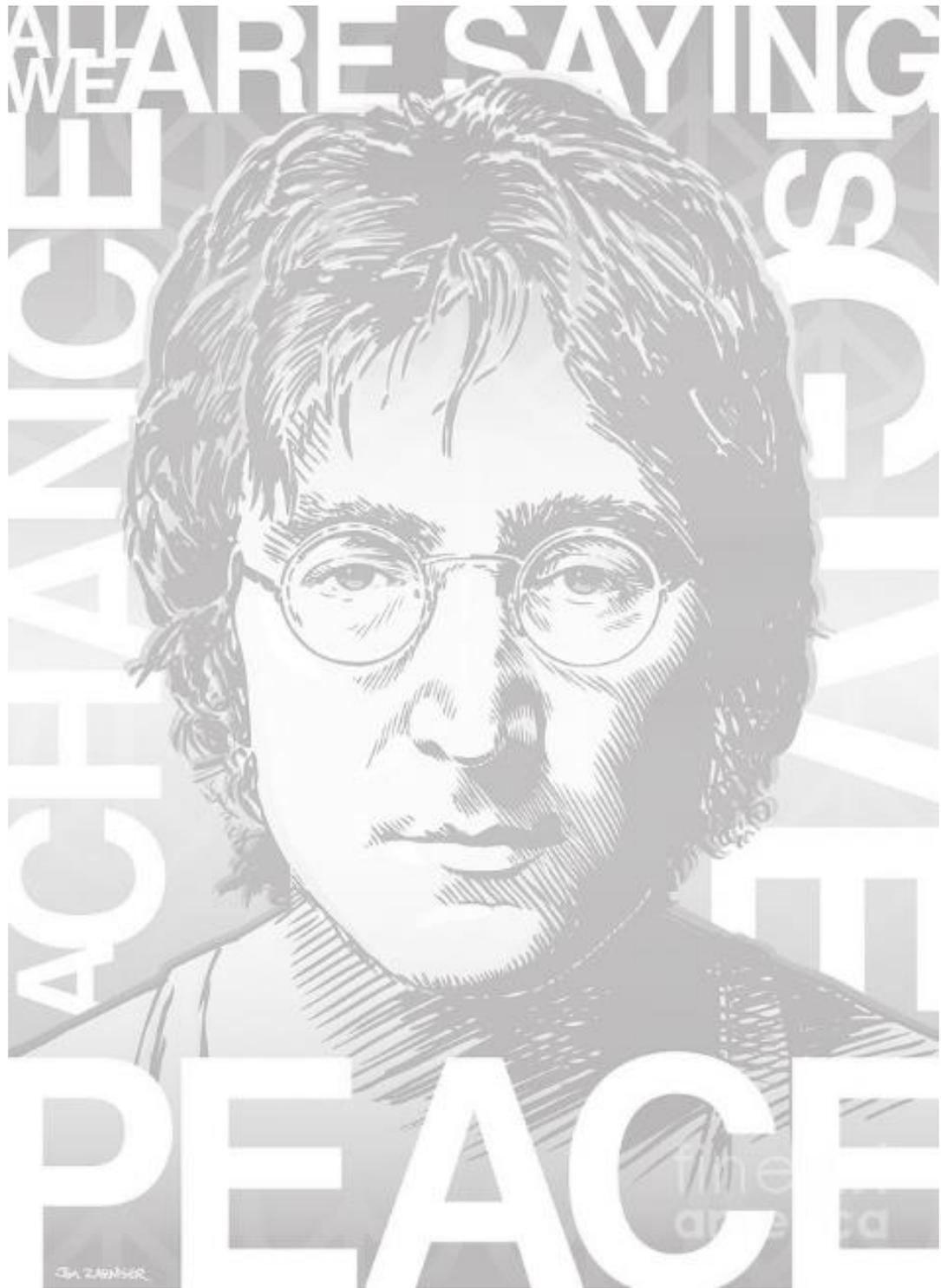
Luis Miguel Herrera Galbán

Tutores: Ing. Breissy Peraza González.

Ing. Yorbenys Pardo Rodríguez.

La Habana, junio de 2016

“Año del 58 Aniversario de la Revolución”



"The more I see, the less I know for sure"

John Lennon.

Declaración de Autoría:

Declaración de Autoría:

Declaramos ser autores de la presente tesis y reconocemos 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 _____.

Autores:

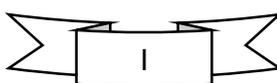
Andy Yonnié Martel Carpio

Luis Miguel Herrera Galbán

Tutores:

Ing. Breissy Peraza González.

Ing. Yorbenys Pardo Rodríguez.



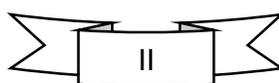
Datos de Contacto:

Ing. Yorbenys Pardo Rodríguez: Graduado de Ingeniero en Ciencias Informáticas en La Universidad de las Ciencias Informáticas en el año 2013. Trabajador del centro GEYSED en el departamento Desarrollo de Componentes, donde desempeña el rol de desarrollador en el proyecto Video-Vigilancia.

Correo Electrónico: ypardor@uci.cu

Ing. Breissy Peraza González: Graduado de Ingeniero en Ciencias Informáticas en La Universidad de las Ciencias Informáticas en el año 2013. Trabajador del centro GEYSED en el departamento Desarrollo de Componentes, donde desempeña el rol de desarrollador en el proyecto Video-Vigilancia.

Correo Electrónico: bperaza@uci.cu



Agradecimientos:

Agradecimientos:

A mis tutores por la ayuda ofrecida. A todos los trabajadores del proyecto de video vigilancia por el apoyo brindado. Al mejor compañero de tesis que me pudo tocar el gran Cuso. A mi novia por todo el amor. A Manuel por ser como un padre para mí. A PapaRafael por ser un ejemplo. A toda mi familia por ser la mejor. A mis amigos Denis y Roniel por todo lo compartido durante estos años. A mis compañeros de la UCI por todo lo vivido. A mis amistades Alejandríto, Piri, Zarza, Toní, Alian, Jorgito, Ale, Luisa, Martica. A mi mejor amiga Ambar y a Tania. A mis vecinos Ale y Anielka por todo lo que los jodo. A mi nueva familia Roxy, Frank y Naura. En fin a todos los que de una manera u otra me apoyaron GRACIAS!!!

Andy.

A mi familia por apoyarme todo este tiempo y darme todo lo que he necesitado.

A mis amigos por estar presentes para mí en todo momento.

A mis tutores por ayudarnos en la realización del documento y la aplicación.

Al tribunal y al oponente por corregirnos los errores y ayudarnos a realizar un mejor documento.

Luís Miguel.



Dedicatoria de Andy:

A mi madre por ser todo para mí, a mi abuela Marta por ser tan especial y a mi tío Frank donde quiera que se encuentre.

Dedicatoria de Luis Miguel:

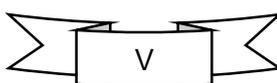
A mis padres, tíos y abuelos que me han apoyado en todo momento.

Resumen:

En la actualidad existen diversas herramientas que mejoran o introducen nuevas funcionalidades en los sistemas de video vigilancia. Entre estas se encuentran los videos sensores, estos permiten detectar automáticamente la ocurrencia de eventos significativos en los videos de los sistemas de videos de vigilancia. El centro GEYSED de la Universidad de las Ciencias Informáticas cuenta con el sistema de video vigilancia Xilema Suria, el cual dispone de diversos videos sensores. La presente solución pone a disposición del operador una herramienta que permite utilizar dichos videos sensores para realizar análisis de videos grabados, además incluye diversas opciones de reproducción especializadas que facilitan el análisis.

Palabras Claves:

análisis de video, eventos significativos, videos sensores.



Abstract:

Currently there are several tools that enhance or introduce new features in video surveillance systems. Among these are video sensors, these allow automatic detection of the occurrence of significant events in surveillance videos. The GEYSED center of the University of Information Science has the video surveillance system Xilema Suria, which has several video sensors. This solution provides the operator with a tool to use these video sensors for analyzing recorded video, it also includes several specialized reproduction options that facilitate the analysis.

Key words:

significant events, video analysis, video sensors.

Índice de contenido:

| | |
|--|-----------|
| INTRODUCCIÓN: | 1 |
| CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA: | 4 |
| 1.1. Introducción | 4 |
| 1.2. Conceptos Asociados al dominio del problema..... | 4 |
| 1.3. Descripción del objeto de estudio | 4 |
| 1.4. Análisis de otras soluciones existentes | 5 |
| 1.5. Análisis, Herramientas y Tecnologías para el desarrollo del software..... | 7 |
| 1.5.1. Metodologías de Desarrollo | 7 |
| 1.5.2. Lenguaje de Modelado Unificado (UML 2.0)..... | 8 |
| 1.5.3. Herramientas CASE..... | 8 |
| 1.5.4. Lenguaje de Programación | 9 |
| 1.5.5. Marco de trabajo | 9 |
| 1.5.6. Entorno de Desarrollo Integrado:..... | 9 |
| 1.5.7. Gestor de Base de Datos: | 10 |
| 1.5.8. Bibliotecas..... | 10 |
| 1.6. Conclusiones Parciales:..... | 12 |
| CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA: | 13 |
| 2.1. Introducción: | 13 |
| 2.2. Modelo de dominio: | 13 |
| 2.2.1. Diagrama de Modelo de dominio..... | 13 |
| 2.2.2. Descripción de las clases del Diagrama del Dominio..... | 13 |
| 2.3. Descripción de la solución | 14 |
| 2.4. Especificación de los requisitos de software | 14 |
| 2.4.1. Requisitos Funcionales | 14 |
| 2.4.2. Requisitos No Funcionales..... | 21 |
| 2.5. Descripción del sistema. | 22 |
| 2.5.1. Definición de los actores | 22 |
| 2.5.2. Listado de los Casos de Uso..... | 22 |

Índice de Contenido:

| | | |
|------------------------------------|--|-----------|
| 2.5.3. | Diagrama de Casos de Uso del Sistema..... | 23 |
| 2.5.4. | Especificación de los Casos de Uso del Sistema | 23 |
| 2.5.5. | Descripción de la Arquitectura..... | 24 |
| 2.6. | Patrones de Diseño: | 26 |
| 2.6.1. | Patrones GRASP | 26 |
| 2.6.2. | Patrones GoF | 27 |
| 2.7. | Modelo de Diseño..... | 28 |
| 2.7.1. | Diagrama de secuencia del diseño | 28 |
| 2.7.2. | Diagrama de clases del diseño | 28 |
| 2.8. | Diagrama Entidad Relación: | 30 |
| 2.9. | Conclusiones parciales:..... | 31 |
| CAPÍTULO 3: | IMPLEMENTACIÓN Y PRUEBAS: | 33 |
| 3.1. | Introducción: | 33 |
| 3.2. | Modelo de Implementación | 33 |
| 3.2.1. | Diagrama de despliegue: | 33 |
| 3.2.2. | Diagrama de Componentes:..... | 33 |
| 3.3. | Pruebas de Software | 34 |
| 3.3.1. | Pruebas de Integración:..... | 35 |
| 3.3.2. | Pruebas del sistema:..... | 35 |
| 3.4. | Conclusiones Parciales:..... | 39 |
| CONCLUSIONES GENERALES: | | 41 |
| RECOMENDACIONES: | | 42 |
| REFERENCIAS BIBLIOGRÁFICAS: | | 43 |
| BIBLIOGRAFÍAS CONSULTADAS: | | 43 |

Índice de Figuras:

| | |
|---|----|
| FIGURA 1. LOGOTIPO <i>LENSEC</i> | 6 |
| FIGURA 2. VISOR DE ARCHIVOS <i>LENSEC</i> | 6 |
| FIGURA 3. LOGOTIPO BLUE IRIS | 6 |
| FIGURA 4. VISOR BLUE IRIS..... | 7 |
| FIGURA 5. DIAGRAMA MODELO DE DOMINIO DEL RECUPERADOR DEL SISTEMA DE VIDEO-VIGILANCIA XILEMA SURIA..... | 13 |
| FIGURA 6. DIAGRAMA DE CASOS DE USO DE LA HERRAMIENTA DE ANÁLISIS DE VIDEOS GRABADOS DEL SISTEMA XILEMA SURIA..... | 23 |
| FIGURA 7. ARQUITECTURA DE LA HERRAMIENTA DE ANÁLISIS DE VIDEOS GRABADOS EN EL SISTEMA DE VIDEO-VIGILANCIA XILEMA SURIA..... | 26 |
| FIGURA 8. DIAGRAMA DE SECUENCIA DE LA FUNCIONALIDAD INICIAR ANÁLISIS. | 28 |
| FIGURA 9. DIAGRAMA DE CLASES DEL CASO DE USO REALIZAR ANÁLISIS..... | 29 |
| FIGURA 10. DIAGRAMA ENTIDAD-RELACIÓN..... | 30 |
| FIGURA 11. DIAGRAMA DE DESPLIEGUE..... | 33 |
| FIGURA 12. DIAGRAMA DE COMPONENTES DE LA HERRAMIENTA DE ANÁLISIS DE VIDEOS GRABADOS EN EL SISTEMA XILEMA SURIA..... | 34 |
| FIGURA 13. RESULTADOS DE LAS PRUEBAS DEL SISTEMA..... | 39 |

Índice de Tablas:

| | | |
|----------|--|----|
| TABLA 1. | ACTORES DEL SISTEMA. | 22 |
| TABLA 2. | CU REALIZAR ANÁLISIS..... | 23 |
| TABLA 3. | DESCRIPCIÓN DE TBL_VIDEO_OFFLINE..... | 30 |
| TABLA 4. | DESCRIPCIÓN DE TBL_IMAGE..... | 30 |
| TABLA 5. | DESCRIPCIÓN DE TBL_TAG..... | 31 |
| TABLA 6. | DESCRIPCIÓN DE TBL_TAG_REMOVE. | 31 |
| TABLA 1. | SC 1 MOSTRAR SENSORES DISPONIBLES. | 36 |
| TABLA 2. | SC 2 SELECCIONAR SENSOR A APLICAR..... | 37 |
| TABLA 3. | SC 3 REALIZAR ANÁLISIS AL VIDEO DEL AR SELECCIONADA APLICANDO EL SENSOR SELECCIONADO. | 37 |
| TABLA 4. | SC 4 CREAR MARCADOR POR ALARMAS GENERADAS. | 38 |

Introducción:

En los últimos años el desarrollo de sistemas de video vigilancia ha sido del interés de investigadores e industrias de todo el mundo. Esto ha traído consigo la constante evolución tecnológica de dichos sistemas, que además de acelerar su proliferación, ha incrementado sustancialmente las funcionalidades de los mismos, especialmente el análisis de video (Lemus, 2013). La video vigilancia ha transcurrido por tres generaciones diferentes, en las que el análisis de video es efectuado de distintas maneras. La primera generación se caracteriza por los sistemas analógicos de circuito cerrado de televisión, que ofrecen escasas herramientas de análisis de video. La segunda introduce los primeros algoritmos de interpretación automática de escenas reales. Ya en la tercera generación los sistemas son completamente digitales, cuentan con cámaras IP que se instalan dentro de una red de datos, lo que ha permitido desarrollar diversas herramientas para el análisis de los videos captados por cámaras de seguridad (M. Valera, 2012).

Entre las principales herramientas para realizar análisis de video se encuentran los video sensores, estos se definen como herramientas de análisis de video digital que analizan secuencias de video y ofrecen información significativa proveniente de las mismas (Robust Motion Detector for Video Surveillance Applications, 2003). Existen diversos tipos de video sensores que permiten detectar un conjunto diferente de eventos, entre los cuáles se pueden mencionar la detección de movimiento y objetos abandonados, control vehicular, reconocimiento facial, entre otros. Las ventajas del uso de video sensores son numerosas. Entre estas se encuentra disminuir el factor del error humano al centrar la atención del operador exclusivamente en los eventos significativos detectados en los videos; reducir las necesidades de almacenamiento, gracias a que le permiten al sistema responder automáticamente a la detección de un evento y permitir la planificación de acciones como la grabación de video.

En Cuba se desarrollan sistemas de video vigilancia que permiten realizar análisis de video, para esto se fomenta el uso de software libre, que constituye una fuente de ahorro para el país por conceptos de licencia. El centro GEYSED perteneciente a la facultad 6 de la Universidad de las Ciencias Informáticas (UCI) cuenta con el sistema de video vigilancia de tercera generación Xilema Suria.

Este está compuesto por los siguientes módulos:

1. Gestor, encargado de la comunicación de los diferentes módulos.
2. Administración, encargado gestionar los roles, usuarios y reglas del sistema
3. Visor, su función es la de visualizar los flujos de video provenientes de las cámaras IP.
4. Grabador, se encarga de grabar los flujos de video provenientes de las cámaras IP.
5. Autonomía, gestiona el espacio de almacenamiento donde se guardan las grabaciones realizadas.
6. Recuperador, encargado de las tareas de búsqueda sobre los videos grabados.

7. Análisis, controla la ejecución en tiempo real de los diferentes video sensores.

A través del módulo de Análisis del sistema Suria se controlan los videos sensores disponibles, estos son activados a petición del usuario desde el Visor y brindan análisis de video solamente en tiempo real sobre los flujos de video provenientes de las cámaras de seguridad instaladas. Para encontrar eventos significativos en los videos grabados se depende totalmente del operador humano, sin que este pueda hacer uso de las potencialidades que ofrecen los videos sensores. Si se tiene en cuenta que según (Albusac, 2015) tras veinte minutos de vigilancia, la atención humana a los detalles del video disminuye considerablemente y además se suman factores como la fatiga producida tras varias horas de trabajo o la dificultad de observar varios videos al mismo tiempo, puede que ocurran muchos eventos significativos que pasen desapercibidos e incluso puede ocurrir que al ser detectados se pierda su ubicación.

A partir de la situación descrita anteriormente, surge el siguiente **problema de investigación**: ¿Cómo analizar los videos grabados por el sistema Xilema Suria para detectar la ocurrencia de posibles eventos significativos? Se plantea como **objeto de estudio**: El análisis de video, enmarcándose en el **campo de acción**: El análisis de videos grabados en el sistema Xilema Suria. Para solucionar el problema planteado se traza como **objetivo general**: Desarrollar una herramienta que analice los videos grabados en el sistema Xilema Suria para detectar la ocurrencia de posibles eventos significativos. Planteando como **idea a defender**: “Con el desarrollo de la herramienta para el análisis de videos grabados se logrará detectar la ocurrencia de posibles eventos significativos”. Se definieron las siguientes **tareas de la investigación**:

1. Caracterizar los procesos relacionados con el análisis de videos grabados.
2. Caracterizar el estado del arte sobre las soluciones existentes que incluyan análisis de videos grabados.
3. Caracterizar técnicas, herramientas y tecnologías a utilizar, para desarrollar la herramienta Análisis de video grabados.
4. Realizar el diseño de la herramienta Análisis de video grabados en el sistema Xilema Suria.
5. Implementar la herramienta Análisis de grabaciones en el sistema Xilema Suria.
6. Validar la solución propuesta.

Para toda investigación es necesario aplicar métodos de investigación que permitan lograr una mayor organización y estructuración del trabajo investigativo. Entre los métodos de investigación se encuentran los métodos teóricos y empíricos. Los **Métodos Teóricos** permiten estudiar las características del objeto de investigación que no son observables directamente (Conferencia Diseño MIC, 2014). En la investigación se utiliza el método **Analítico-Sintético**, mediante el cual se realiza un análisis de la bibliografía a consultar y se extrae lo esencial de la misma (Sampieri, 2010), también se utiliza el método **Histórico-Lógico** para analizar la evolución de los sistemas de video-vigilancia (Sampieri, 2010).

Por otra parte están los **Métodos Empíricos** que representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional (Conferencia Diseño MIC, 2014), entre estos está la **Observación**, esta permite conocer la realidad mediante la percepción directa de los objetos y fenómenos, la utilización de la misma facilita la familiarización con el trabajo dentro del proyecto de Video Vigilancia Xilema Suria (Sabino, 2014), también se utiliza la **Tormenta de Ideas**, esta es una técnica de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios. Su utilización permite generar una gran variedad de vistas del problema, sobre todo al comienzo del proceso de captura de requisitos (Sabino, 2014).

El presente trabajo está distribuido en los siguientes capítulos:

Capítulo 1: Fundamentación teórica. En este capítulo se definen los conceptos asociados a la investigación. Se realiza un análisis del objeto de estudio y estado del arte sobre soluciones existentes. Además se caracterizan y seleccionan las tecnologías, el lenguaje de programación, las herramientas y la metodología para el desarrollo de la solución.

Capítulo 2: Análisis y diseño del sistema. En este capítulo se describe el modelo del dominio del sistema que se va a desarrollar. También se identifican los requisitos funcionales y no funcionales. Se definen los casos de usos y actores del sistema. Se especifica la arquitectura de la aplicación.

Capítulo 3: Implementación y prueba. En este capítulo se muestran los artefactos generados en la implementación, basados en la metodología utilizada. También se realizan las descripciones pertinentes al proceso de pruebas.

Capítulo 1: Fundamentación Teórica:

1.1. Introducción

En este capítulo se presentan los principales conceptos relacionados con el análisis de video. Se muestra el estudio realizado a varios sistemas que incluyen en su solución el análisis de videos grabados. Se lleva a cabo una caracterización de las herramientas y tecnologías a utilizar.

1.2. Conceptos Asociados al dominio del problema

A continuación se muestran una serie de conceptos asociados al dominio del problema para un mayor entendimiento del mismo.

Video: El término video es la tecnología de la grabación, almacenamiento o transmisión de imágenes por medios electrónicos digitales o analógicos de una secuencia de imágenes que representan escenas en movimiento acompañadas o no por sonido (Real Academia Española, 2016). En los sistemas de video vigilancia el término video es tratado de dos formas diferentes, una se refiere al flujo de imágenes en tiempo real provenientes de cámaras de seguridad activas en el sistema, y la otra a los archivos videos grabados en el sistema.

Evento Significativo: No existe un concepto único que enmarque evento significativo como un solo concepto en las diferentes bibliografías consultadas, por lo que se hace necesario analizar evento y significativo por separado, para poder arribar al concepto tratado en la investigación. Evento se puede definir como cualquier acontecimiento, circunstancia, suceso o incidente programado o no (Real Academia Española, 2016). Significativo hace referencia a que tiene importancia por representar o significar algo (Real Academia Española, 2016). Por lo que evento significativo es tratado como un suceso o incidente de importancia en los videos de las cámaras de seguridad.

1.3. Descripción del objeto de estudio

El análisis de video es tratado de diferentes formas en las diversas bibliografías consultadas sin llegar a un concepto único que enmarque ambos términos. Para una mejor comprensión del término se hace necesario un desglose de ambos conceptos. Análisis hace referencia a la distinción y separación de las partes de algo para conocer su composición (OXFORD University Press, 2016), también se puede definir como un estudio detallado de algo (Real Academia Española, 2016). Si se toma el concepto de análisis y video anteriormente planteado, el término análisis de video en el marco de esta investigación es tratado como las operaciones realizadas para detectar la ocurrencia de eventos significativos tanto por el ordenador, a través de video sensores, como por el operador humano que visualiza los videos.

La visualización de videos por parte del operador humano fue el primer tipo de análisis de video que permitían los sistemas de video vigilancia, a través de estaciones de monitoreo o reproductores especializados. Luego fue incorporado el análisis por el ordenador, gracias a las funcionalidades que

ofrecen los videos sensores, en varias de las bibliografías consultadas este tipo de análisis se denomina análisis de video inteligente. Según (G. Medioni, 2001) se puede describir como un amplio número de nuevas tecnologías y evoluciones en el campo de la video vigilancia que van desde la capacidad del sistema de detectar movimiento, hasta otras que analizan imágenes de forma similar a como lo haría una persona. De manera general para la implementación de un video sensor que permita análisis de video inteligente intervienen las siguientes fases:

1. Obtención de *frames* o imágenes.
2. Pre procesado de la imagen para mejorar su calidad.
3. Segmentación, en esta se definen los objetos que se encuentran dentro de la imagen.
4. Representar las regiones que son de interés.
5. Reconocimiento y Descripción de los objetos presentes.
6. Interpretación de la información obtenida.

La respuesta del sistema tras la ejecución de la última fase es configurable y varía según el tipo de video sensor.

Los video sensores poseen la capacidad de ejecutar las distintas fases en tiempo real, estos obtienen y procesan las imágenes captadas por cámaras de seguridad activas, gracias a esto se han podido desarrollar estaciones de monitoreo que interpretan sin “depender del ojo humano” situaciones de riesgo. Lo que permite planificar acciones o generar alertas al detectar eventos significativos (GsSeguridad, 2016). Es posible también realizar análisis de video inteligente si las imágenes de entrada a los videos sensores provienen de videos grabados, gracias a esto se han desarrollado reproductores que se especializan en el análisis de videos grabados, en los cuáles se visualizan marcadores por eventos detectados. Lo que permite concentrar la atención del operador exclusivamente en información relevante y reducir el factor de error humano. Los reproductores actuales para optimizar la visualización de los videos grabados ofrecen distintas funcionalidades como (Alava, 2015):

- Visualizar varios videos a la vez.
- Realizar anotaciones en instantes específicos.
- Capturar fotogramas relevantes.
- Controlar parámetros de imagen como el brillo y el contraste.

1.4. Análisis de otras soluciones existentes

En la actualidad existen numerosas empresas e instituciones que desarrollan sistemas de video vigilancia. Entre las que incorporan análisis de video inteligente se encuentran:

Perspective VMS, Lensec:



Figura 1. Logotipo Lensec

Perspective VMS Lensec es un sistema de video vigilancia que cuenta con video sensores de detección de movimiento y detección de objetos abandonados. Este sistema está dividido en módulos entre los cuales se encuentra el Visor de Archivos, el cual apoyado en análisis de video inteligente proporcionado por su gama de video sensores, destaca los intervalos de tiempo en que fueron detectados eventos significativos. Además de facilitar la visualización, permite manipular simultáneamente hasta un máximo de tres videos grabados. También permite al usuario etiquetar eventos que el mismo desee destacar. Las operaciones que se pueden realizar en su reproductor son bastante básicas como la reproducción y pausa del video, encolar videos y descargar el video.

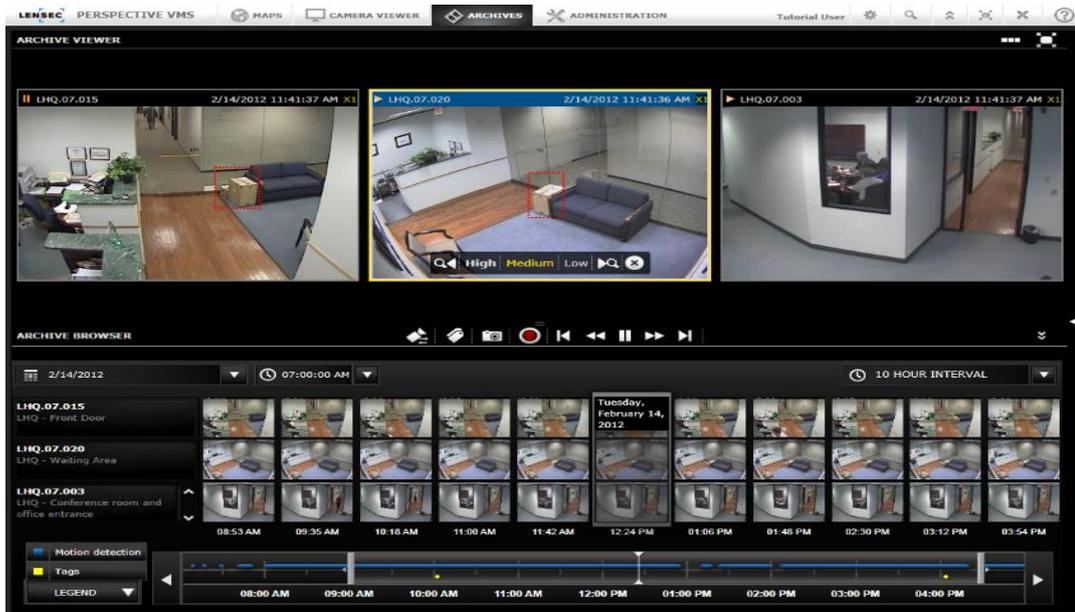


Figura 2. Visor de Archivos Lensec.

Blue Iris:



Figura 3. Logotipo Blue Iris

Es un sistema de video vigilancia que cuenta con video sensores de detección de movimiento y audio. Estos son integrados al sistema a través del visor de videos grabados donde se muestra el instante de

Fundamentación Teórica:

video donde se activó algún video sensor e indica fecha, hora y motivo de la alarma. En este se permiten las operaciones de reproducción y pausa de videos, encolar los mismos, controlar parámetros de las imágenes como por ejemplo el brillo. Además de permitir tomar y mostrar instantáneas que el operador quiera destacar.



Figura 4. Visor Blue Iris

Ambas herramientas integran análisis de video en su solución, por lo que su estudio sirve de punto de partida para conocer ejemplos reales de software que implementan análisis de video. Ninguna de las herramientas analizadas puede adoptarse como solución para el problema inicial, al no existir alguna que implemente todas las funcionalidades necesarias, además de que ambas herramientas no se desarrollaron con las mismas tecnologías que el Sistema Xilema Suria.

1.5. Análisis, Herramientas y Tecnologías para el desarrollo del software.

1.5.1. Metodologías de Desarrollo

Según (Pressman, 2010) las metodologías de desarrollo de software son el marco de trabajo que colecciona un conjunto de pasos y procedimientos que se deben seguir para organizar, controlar y planear el proceso de desarrollo de un software. Existen dos tipos de metodologías, las ágiles y las

pesadas. En el proyecto de video vigilancia Xilema Suria la metodología se encuentra definida, se utiliza la variación realizada en la UCI del “Proceso Unificado Ágil” (AUP) por sus siglas en inglés.

Proceso Unificado Ágil variación UCI (AUP UCI):

AUP es una versión simplificada del Proceso Unificado Racional (RUP) por sus siglas en inglés. Su uso proporciona un enfoque simple, fácil de entender, del desarrollo de software. Utiliza técnicas y conceptos ágiles válidos en RUP, como por ejemplo el desarrollo orientado a pruebas, modelado ágil, gestión de cambios ágil y refactorización de base de datos (Abdolazimian, y otros, 2008).

Las organizaciones que utilizan AUP incorporan sus propias modificaciones con tal de ajustarla a sus necesidades. En esta investigación se selecciona la versión UCI de AUP. La cuál propone las siguientes disciplinas: Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas Internas, Pruebas de liberación, Pruebas de aceptación, Gestión de configuración, Gestión de proyecto y Entorno. Se selecciona el escenario “Proyectos que modelen el negocio con modelo conceptual solo pueden modelar el sistema con casos de usos” este permite modelar solo los conceptos fundamentales del sistema a través de casos de uso. El uso de AUP supone varias ventajas, entre las que se encuentran que el personal en todo momento sabe lo que se hace sin conocer los detalles específicos que se encuentran en la documentación, simplicidad gracias a que se describen las fases de forma concisa, agilidad obtenida de los procesos ágiles de RUP, centrarse en actividades de alto valor enfocados en el resultado final, recomienda el uso de herramientas que mejor se adapten al trabajo y es fácil la adaptación de este producto para que cumpla con las necesidades propias (Rusu, y otros, 2011).

1.5.2. Lenguaje de Modelado Unificado (UML 2.0).

Según (Jacobson, 2005) El Lenguaje de Modelado Unificado (UML) es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. Su uso se debe a que este ofrece un estándar para describir el modelo del sistema; permite representar aspectos conceptuales tales como procesos de negocios, funciones del sistema, esquemas de base de datos y componentes de software.

1.5.3. Herramientas CASE

Las herramientas de Ingeniería de Software Asistida por Computadora (CASE por sus siglas en inglés), son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante los pasos del ciclo de vida de desarrollo de un software. Las herramientas CASE también permiten a los analistas tener más tiempo para el análisis y diseño, así como disminuir el tiempo para codificar y probar (Case, 2001).

Visual Paradigm 8.0

Visual Paradigm es una herramienta CASE. Su uso propicia un conjunto de ayudas para el desarrollo del software propuesto en esta investigación, que van desde la planificación, el análisis y diseño, hasta la generación del código fuente de los programas y la documentación.

1.5.4. Lenguaje de Programación

C++:

C++ es un lenguaje imperativo orientado a objetos derivado del C. Es una mejoría sobre muchas de las características de C, y proporciona capacidades de P.O.O. (Programación Orientada a Objeto) que aporta mucho para incrementar la productividad, calidad y reutilización del software. Este lenguaje es muy eficaz en cuanto a rapidez y uso de memoria en las aplicaciones que se obtienen. Las bibliotecas estándar de C++ proporcionan un conjunto extenso de capacidades de entrada/salida. En el diseño de C++ prima sobre todo la velocidad de ejecución del código (Josutis, 2012).

Se selecciona este lenguaje por ser de uno de los más rápidos en cuanto a ejecución por lo que facilita y ayuda en el trabajo con video, además el sistema Suria está desarrollado con este lenguaje.

1.5.5. Marco de trabajo

Según (Quesada, 2008) un marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que permite resolver nuevos problemas. En el desarrollo de software constituye una estructura conceptual y tecnológica de soporte definida. Normalmente, cuenta con artefactos o módulos de software concretos, que se emplean fundamentalmente para desarrollar código fuente.

Qt 5.3

Se seleccionó este marco de trabajo ya que es multiplataforma, de código abierto y se utiliza generalmente para desarrollar aplicaciones de software que posean interfaz gráfica. Además cuenta con soporte para la comunicación con bases de datos y numerosas bibliotecas que facilitan el trabajo con archivos multimedia. Qt utiliza C++ como lenguaje de programación nativo (Rischpater, 2013).

1.5.6. Entorno de Desarrollo Integrado:

Un Entorno de Desarrollo Integrado (por sus siglas en inglés, IDE) es un programa que está compuesto por un conjunto de herramientas para un programador. Facilita un marco de trabajo amigable para una gran cantidad de lenguajes de programación tales como C++, Java y C#, también permite utilizar uno o varios lenguajes de programación (Fritzler, 2011).

Qt Creator 3.1:

Se selecciona Qt *Creator* como IDE ya que este es multiplataforma y permite desarrollar aplicaciones en C++ de manera sencilla y rápida. Además como su nombre lo indica, está basado en la biblioteca Qt y cuenta con un editor avanzado para C++, con diseñador de formularios (GUI) integrado, con

herramientas para la administración y construcción de proyectos, con completado automático y un depurador visual (Rischpater, 2013).

Qt *Creator* cumple con las políticas del centro GEYSED y es utilizado en el proyecto SURIA, factor que favorece su selección.

1.5.7. Gestor de Base de Datos:

Un Sistema Gestor de Base de Datos (SGBD) es un sistema de software que permite el almacenamiento, modificación y extracción de la información en una base de datos, ya sea de forma interactiva o a través de un lenguaje de programación. (Bertino, y otros, 1995).

PostgreSQL 9.3:

Para la implementación de la herramienta se utilizara *PostgreSQL* en su versión 9.3. Este es un potente Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos. Entre las características que potenciaron su elección se encuentran que está distribuido bajo la licencia BSD (*Berkeley Software Distribution*), su código está disponible libremente, cuenta con una amplia comunidad de desarrollo, es multiplataforma, soporta casi toda la sintaxis SQL y permite el enlace con muchos lenguajes de programación. (PostgreSQL, 2016).

Administrador de Base Datos (*pgAdmin 3*):

Para la administración de base de datos de *PostgreSQL* se utilizará el *pgAdmin 3*, este es una plataforma de código abierto desarrollada por una comunidad de expertos disponible en varios idiomas. Su uso se debe al estar publicada bajo la licencia de BSD, contar con una interfaz gráfica compatible con todas las características de *PostgreSQL*, además está disponible en diversas plataformas. (pgAdmin, 2016)

1.5.8. Bibliotecas

Las bibliotecas son un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca. Estas facilitan la implementación de muchas de las funcionalidades de la herramienta. (Prezi, 2016)

Biblioteca *OpenCV 2.4.9*

Se selecciona la biblioteca *OpenCV 2.4.9* (*Open Source Computer Vision Library*) ya que esta proporciona un alto nivel de funciones para el procesamiento de imágenes. Es de código abierto, gratuita, multiplataforma, está desarrollada bajo la licencia BSD y es de fácil uso (OpenCV, 2016). Los videos sensores disponibles en el sistema Xilema Suria utilizan esta biblioteca que es utilizada para realizar las modificaciones a los sensores y notificar el instante de detección de los eventos significativos.

Biblioteca *libvlc 2.2.2*

Fundamentación Teórica:

La biblioteca *libvlc* es una interfaz de programación de aplicaciones (API por sus siglas en inglés). Se selecciona al poseer una amplia gama de funcionalidades que pueden ser utilizadas de forma gratuita y permiten realizar diversas operaciones sobre archivos de media (VideoLan, 2016), esta facilita la implementación de diversas funcionalidades relacionados con el reproductor que incluye la herramienta de análisis de videos grabados del sistema Xilema Suria.

1.6. Conclusiones Parciales:

En este capítulo se definieron los puntos relacionados con la fundamentación teórica de la investigación, Arribándose a las siguientes conclusiones:

- ✓ La inclusión de video sensores en sistemas de video-vigilancia permiten la detección automática de eventos significativos en videos grabados, para evitar que estos pasen desapercibidos.
- ✓ Existen diversos sistemas de video vigilancia que incluyen funcionalidades que permiten el análisis de videos grabados, pero ninguno implementa todas las funcionalidades necesarias. Por lo que no pueden ser adoptados como solución al problema de investigación planteado.
- ✓ Las herramientas y tecnologías seleccionadas fueron definidas en el centro de desarrollo GEYSED. Además de las ventajas planteadas en el capítulo del uso de las mismas, estas favorecen a la integración de la herramienta análisis de videos grabados con el sistema de video vigilancia Xilema Suria.

Capítulo 2: Análisis y Diseño del Sistema:

2.1. Introducción:

En el presente capítulo se realiza un estudio de las principales características del sistema, para facilitar su posterior implementación. Se identifican los requisitos funcionales y no funcionales, se agrupan los primeros en casos de uso y se presenta una descripción de cada uno de ellos. También se presentan los diagramas de clases del diseño y otros artefactos relacionados con la metodología AUP UCI.

2.2. Modelo de dominio:

El modelo de dominio permite de manera visual mostrar al usuario los principales conceptos asociados a la situación problemática planteada. Es una representación de las clases conceptuales del mundo real, no de componentes de software y emplea un glosario de términos para lograr una mejor representación de los conceptos asociados (Larman, 2003).

2.2.1. Diagrama de Modelo de dominio

En la figura 5 se muestra la representación del modelo de dominio.

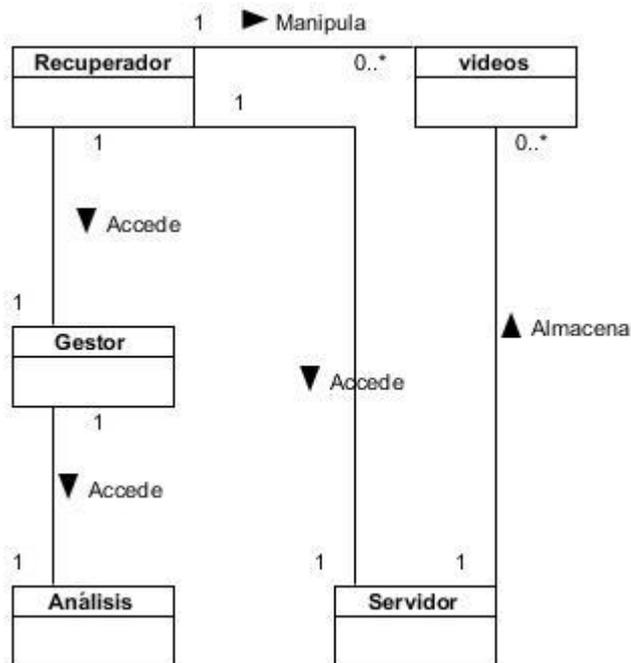


Figura 5. Diagrama Modelo de Dominio del Recuperador del sistema de video-vigilancia Xilema Suria.

2.2.2. Descripción de las clases del Diagrama del Dominio

Videos: Son los videos grabados en el sistema Xilema Suria y son almacenados en el servidor.

Recuperador: Módulo del sistema Xilema Suria que permite acceder a los videos que están almacenados en el servidor.

Servidor: Es donde se almacenan los videos capturados por el sistema Xilema Suria.

Gestor: Módulo del sistema Xilema Suria que permite la comunicación entre los diferentes módulos del sistema.

Análisis: Módulo del sistema Xilema Suria que permite controlar la ejecución en tiempo real de los diferentes video sensores.

El recuperador accede al Gestor para obtener la dirección de los videos grabados, luego accede al servidor donde se almacenan los videos para recuperarlos. El Gestor accede al Análisis para permitir analizar los flujos de videos provenientes de las cámaras del sistema.

2.3. Descripción de la solución

La herramienta análisis de videos grabados del sistema Xilema Suria se valdrá del módulo Análisis y los videos sensores disponibles en él para realizar el procesamiento inteligente de video. La misma utilizará la respuesta de estos para mostrar en la línea de tiempo los marcadores detectados por cada evento significativo de acuerdo al video seleccionado, además permitirá al operador crear marcadores en instantes de tiempo que considere significativos. Contará de un reproductor que permita manipular de uno a cuatro videos a la vez, presentará opciones para encolar videos, controlar parámetros en los mismos como el brillo, contraste, volumen de audio, zoom, velocidad de reproducción y descarga del mismo. Además permitirá el trabajo con instantáneas, por lo que ofrecerá opciones para la captura y visualización de las mismas.

2.4. Especificación de los requisitos de software

La especificación de requisitos software no es más que la descripción completa del comportamiento del sistema que se desea desarrollar (Pressman, 2010). Estos ofrecen con claridad las necesidades tanto de los usuarios como las del sistema.

2.4.1. Requisitos Funcionales

Los requisitos funcionales (RF) son declaraciones de servicios que el sistema debe proporcionar, definen la manera en que debe reaccionar a determinadas entradas y cómo se debe comportar en situaciones particulares (Pressman, 2010). A continuación se enumeran los RF y se describen brevemente.

RF1. Mostrar áreas de reproducción(AR):

El Sistema debe mostrar las áreas donde serán analizados los videos seleccionados y el área de visualización de instantáneas.

Salidas: De 1 hasta 4 AR.

RF2. Seleccionar AR:

El Sistema debe permitir seleccionar un AR, la cual pasará a ser manipulada por los controles del sistema y se actualiza la línea de tiempo con marcadores del video del AR seleccionada.

Salida: AR seleccionada.

Salida: video en reproducción del área correspondiente.

Salida: línea de tiempo actualizada con sus marcadores.

RF3. Visualizar línea de tiempo y controles:

El Sistema debe mostrar la línea de tiempo y los controles de la reproducción.

Salidas: línea de tiempo.

RF4. Seleccionar videos recuperados:

El Sistema debe permitir seleccionar los videos que van a ser analizados en cada AR.

Entrada: lista de videos.

Entrada: AR seleccionada.

Salida: videos seleccionados.

RF5. Crear lista de reproducción:

El Sistema debe permitir crear una lista de reproducción con los videos que van a ser analizados en cada AR.

Entrada: lista de videos.

Entrada: AR seleccionada.

Salida: lista de reproducción.

RF6. Posicionarse en un instante de tiempo del video:

El Sistema debe permitir reproducir el video del AR seleccionada a partir del instante de tiempo que desee el usuario.

Entrada: AR seleccionada

Entrada: instante de tiempo.

Salida: video del AR reproducido a partir del instante de tiempo seleccionado.

RF7. Iniciar reproducción del video:

El Sistema debe permitir iniciar la reproducción del video cuando el usuario lo solicite:

Entrada: AR seleccionada.

Salida: video del AR seleccionada reproduciéndose.

RF8. Pausar reproducción del video:

El Sistema debe permitir pausar la reproducción del video cuando el usuario lo solicite.

Entrada: AR seleccionada.

Salida: video del AR seleccionada pausado.

RF9. Disminuir la velocidad de reproducción del video:

El Sistema debe permitir disminuir la velocidad de reproducción del video que se encuentra en el AR seleccionada.

Entrada: AR seleccionada.

Salida: video del AR seleccionada con la velocidad de reproducción actualizada.

RF10. Acelerar la velocidad de reproducción del video:

El Sistema debe permitir acelerar la velocidad de reproducción del video del AR seleccionada.

Entrada: AR seleccionada.

Salida: video del AR seleccionada con la velocidad de reproducción actualizada.

RF11. Adelantar al video siguiente en la lista de reproducción:

El Sistema debe permitir pasar al siguiente video en la lista de reproducción del AR seleccionada.

Entrada: lista de reproducción del AR seleccionada.

Salida: próximo video.

RF12. Retroceder al video anterior en la lista de reproducción:

El Sistema debe permitir regresar al video anterior en la lista de reproducción.

Entrada: lista de reproducción del AR seleccionada.

Salida: video anterior.

RF13. Visualizar en modo pantalla completa.

El sistema debe permitir visualizar el video del AR seleccionada en modo pantalla completa y volver a su modo normal.

Entrada: AR seleccionada.

Salida: imagen del video del AR visualizándose en modo de pantalla completa.

RF14. Controlar volumen:

El sistema debe permitir controlar el volumen de audio del video del AR seleccionada.

Entrada: AR seleccionada.

Entrada: volumen de audio definido por el usuario.

Salida: volumen de audio del video del AR seleccionada actualizado, con el volumen definido por el usuario.

RF15. Establecer brillo:

El sistema debe permitir controlar el brillo de imagen del video del AR seleccionada.

Entrada: AR seleccionada.

Entrada: brillo de imagen definido por el usuario.

Salida: brillo de imagen del video del AR seleccionada actualizado, con el brillo definido por el usuario.

RF16. Establecer contraste:

El sistema debe permitir controlar el contraste de imagen del video del AR seleccionada.

Entrada: AR seleccionada.

Entrada: contraste de imagen definido por el usuario.

Salida: contraste de imagen del video del AR seleccionada actualizada, con el contraste definido por el usuario.

RF17. Hacer Zoom:

El sistema debe permitir controlar el tamaño de imagen del video del AR seleccionada.

Entrada: AR seleccionada.

Entrada: tamaño de imagen definido por el usuario.

Salida: tamaño de imagen del video del AR seleccionada actualizada, con el tamaño definido por el usuario.

RF18. Realizar anotaciones al video del AR seleccionada:

El Sistema debe permitir realizar anotaciones al video del AR seleccionada.

Entrada: AR seleccionada.

Entrada: texto de la anotación.

Salida: anotación.

RF19. Modificar anotaciones al video del AR seleccionada:

El Sistema debe permitir modificar anotaciones al video del AR seleccionada.

Entrada: AR seleccionada.

Entrada: texto de la anotación.

Salida: anotación modificada.

RF20. Eliminar anotaciones al video del AR seleccionada:

El Sistema debe permitir eliminar anotaciones al video del AR seleccionada.

Entrada: AR seleccionada.

RF21. Mostrar anotaciones al video del AR seleccionada:

El Sistema debe permitir mostrar anotaciones al video del AR seleccionada.

Entrada: AR seleccionada.

Salida: anotación.

RF22. Mostrar sensores disponibles:

El Sistema debe permitir visualizar los sensores que están disponibles para aplicarse al video.

Salida: sensores disponibles.

RF23. Seleccionar sensor a aplicar:

El Sistema debe permitir seleccionar el sensor que se va a aplicar.

Entrada: sensor.

Salida: sensor seleccionado.

RF24. Realizar análisis al video del AR seleccionada aplicando el sensor seleccionado:

Entrada: AR seleccionada.

Entrada: sensor seleccionado.

Salida: línea de tiempo con marcadores actualizada.

RF25. Crear marcador por alarmas generadas:

El Sistema debe permitir crear un marcador cada vez que ocurra un evento significativo.

Entrada: AR seleccionada.

Salida: línea de tiempo con marcadores actualizada.

RF26. Visualizar línea de tiempo y áreas de marcadores:

El Sistema debe mostrar la línea de tiempo y las áreas de marcadores.

Salida: línea de tiempo.

Salida: áreas de marcadores.

RF27. Mostrar marcadores en las áreas de marcadores:

El Sistema debe permitir visualizar en las áreas de marcadores, los marcadores pertenecientes al video del AR seleccionada.

Entrada: AR seleccionada.

Salida: línea de tiempo con marcadores.

RF28. Insertar marcadores de forma manual:

El Sistema debe permitir que el usuario inserte marcadores en la línea de tiempo del video del AR seleccionada.

Entrada: AR seleccionada.

Salida: nuevo marcador.

Salida: línea de tiempo actualizada con el nuevo marcador.

RF29. Modificar metadato del marcador:

El Sistema debe permitir modificar el metadato del marcador que ha sido seleccionado.

Entrada: marcador seleccionado.

Entrada: texto del marcador seleccionado.

Salida: marcador actualizado.

RF30. Eliminar marcadores:

El Sistema debe permitir eliminar marcadores que han sido previamente insertados.

Entrada: marcador a eliminar.

Salida: línea de tiempo actualizada.

RF31. Recuperar marcadores:

El Sistema debe permitir recuperar marcadores que han sido eliminados.

Salida: línea de tiempo con los marcadores que fueron recuperados.

RF32. Mostrar instantánea en miniatura cuando se posicione el cursor en un instante de tiempo:

El Sistema debe permitir visualizar una miniatura de la instantánea correspondiente al instante de tiempo sobre el que está posicionado el cursor.

Entrada: AR seleccionada:

Salida: miniatura de la instantánea.

RF33. Guardar instantánea:

El Sistema debe permitir guardar la instantánea correspondiente a un instante de tiempo del video seleccionado.

Entrada: video seleccionado.

Salida: instantánea del video almacenada.

RF34. Visualizar instantáneas:

El Sistema debe permitir visualizar las instantáneas correspondientes al video seleccionado.

Entrada: AR seleccionada.

Salida: instantáneas del video del AR seleccionada.

RF35. Dividir un archivo en instantáneas.

El Sistema debe permitir establecer la cadencia para obtener y guardar instantáneas correspondientes al video del AR seleccionada:

Entrada: AR seleccionada.

Entrada: cadencia.

Salida: instantáneas.

RF36. Reproducir a partir del instante de tiempo correspondiente a la instantánea seleccionada:

El Sistema debe permitir trasladarse al instante que corresponde la instantánea seleccionada.

Entrada: AR seleccionada.

Entrada: instantánea.

Salida: video del AR posicionado en el instante de tiempo correspondiente a la instantánea.

RF37. Convertir instantánea en marcador:

El sistema debe permitir convertir una instantánea en marcador.

Entrada: instantánea.

Salida: marcador.

RF38. Reproducir a partir del instante de tiempo correspondiente al marcador seleccionado:

El Sistema debe permitir trasladarse al instante que corresponde el marcador seleccionado.

Entrada: AR seleccionada.

Entrada: marcador.

Salida: video del AR posicionado en el instante de tiempo correspondiente al marcador.

RF39. Descargar videos recuperados:

El Sistema debe permitir descargar el video seleccionado y poder decidir la dirección donde será guardado, además se debe visualizar el progreso de la descarga.

Entrada: AR seleccionada.

Entrada: Dirección donde se desea guardar el video.

Salida: video descargado en la dirección especificada.

2.4.2. Requisitos No Funcionales

Según (Pressman, 2010) los requisitos no funcionales son las características que hacen a un sistema atractivo, usable, rápido o confiable. Se definieron los siguientes requisitos no funcionales.

Requisitos de Usabilidad:

- RNF1. El sistema permitirá ser configurable para lograr la comodidad del usuario.
- RNF2. El sistema deberá presentar una interfaz intuitiva que facilite la interacción con las funcionalidades que brinda.

Requisitos de Software:

- RNF1. Sistema Operativo Debian 8
- RNF2. Biblioteca *libvlc* 2.2.2.
- RNF3. Biblioteca *OpenCV* 2.4.9.

Rendimiento

- RNF4. El máximo de memoria RAM que debe de consumir el sistema es de 200 Mb.

Disponibilidad

- RNF5. El componente puede estar funcionando a 24 horas, tiempo completo.

Hardware

- RNF6. Se debe tener como mínimo 1Gb de RAM y procesador i3.

Restricciones del diseño

- RNF7. El lenguaje que se utilizará para el desarrollo del sistema será C++.

2.5. Descripción del sistema.

2.5.1. Definición de los actores

Un actor del sistema no es más que un conjunto de roles que los usuarios desempeñan cuando interactúan con los casos de uso (Pressman, 2010).

Tabla 1. Actores del sistema.

| Actor | Descripción |
|----------|--|
| Operador | Es el encargado de realizar el análisis de los videos, realiza operaciones con el reproductor de video, administra las instantáneas, además gestiona los marcadores y notas. |

2.5.2. Listado de los Casos de Uso.

Un Caso de Uso (CU) es un fragmento de una funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores, el cual especifica una secuencia de acciones o un determinado comportamiento que el sistema puede llevar a cabo durante la interacción con los actores (Jacobson, y

otros, 2000). De acuerdo a los RF definidos anteriormente, se agruparon en los siguientes casos de usos:

1. Tratar áreas de reproducción.
2. Realizar operaciones con el reproductor.
3. Realizar análisis.
4. Gestionar nota.
5. Gestionar marcador.
6. Administrar instantánea.

2.5.3. Diagrama de Casos de Uso del Sistema

Un Diagrama de Casos de Uso del Sistema representa gráficamente a los procesos y sus interacciones con los actores (Jacobson, y otros, 2000). En la figura 6 se muestra el diagrama de Casos de uso del Sistema.

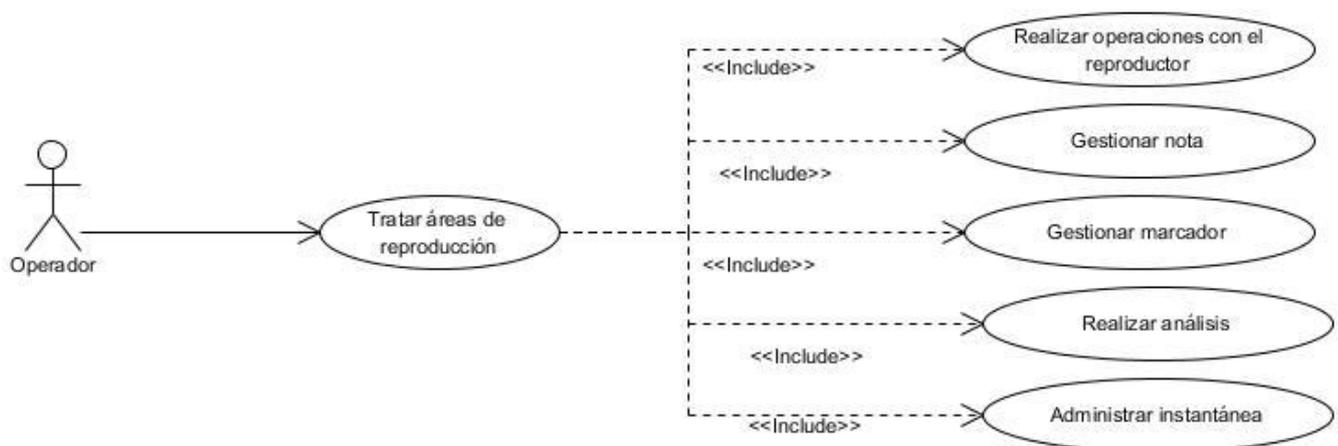


Figura 6. Diagrama de casos de uso de la herramienta de análisis de videos grabados del sistema Xilema Suria.

2.5.4. Especificación de los Casos de Uso del Sistema

Tabla 2. CU Realizar Análisis.

| CU3 | Realizar Análisis. |
|-----------------------|---|
| Objetivo | Aplicar video sensores al video seleccionado. |
| Actores | Operador. |
| Resumen | Este caso de uso permite aplicar los sensores al video seleccionado y adicionar marcadores cuando los sensores detecten que ocurrió un evento no deseado. |
| Complejidad | Alta. |
| Prioridad | Crítica. |
| Precondiciones | El video a analizar debe haber sido recuperado. |

Análisis y Diseño del Sistema:

| | | |
|--|---|--|
| | Se debe seleccionar el área de reproducción. Debe haber video en el área de reproducción seleccionada. | |
| Postcondiciones | Video Recuperado | |
| Flujo de eventos | | |
| Flujo básico: Realizar análisis. | | |
| | Actor | Sistema |
| 1. | Selecciona la opción mostrar sensores. | |
| 2. | | Muestra los sensores disponibles. |
| 3. | Selecciona el sensor que se va a aplicar y selecciona aceptar. | |
| 4. | | Analiza los videos a partir de los sensores. |
| 5. | | Inserta un marcador donde ocurrió el evento. |
| 6. | | Muestra la línea de tiempo con los marcadores. Termina el CU. |
| Flujos alternos. | | |
| 3a. Cancela la operación de seleccionar el sensor. | | |
| | Actor | Sistema |
| | | Cierra la ventana. Regresar al paso 1. |
| Relaciones | CU Incluidos | No procede. |
| | CU Extendidos | No procede. |
| Referencias | RF22 RF23 RF24 RF25 RF27 | |

2.5.5. Descripción de la Arquitectura.

La Arquitectura de Software se refiere a “La organización fundamental de un sistema encarnada en sus componentes y las relaciones entre estos, además son los principios que orientan el diseño y evolución del mismo” (Bass, 2013). A continuación se analiza la arquitectura general del sistema y la del módulo Recuperador para la comprensión de las mismas. Además será definida la arquitectura que se utilizará para implementar la herramienta de análisis de videos grabados.

Arquitectura en Pizarra:

La Arquitectura en Pizarra consiste en desarrollar el sistema sobre la base de componentes que son llamados agentes autónomos, estos realizan todas sus funcionalidades de forma independiente. Pero a su vez dependen de la entrada de información externa es decir de información que necesitan y solicitan

para su correcto funcionamiento, que es provista por un elemento central, comúnmente denominado Repositorio Activo de información (Reynoso, 2004).

El sistema Xilema Suria implementa esta arquitectura, en la cual el Gestor es el Repositorio Activo, y los demás módulos del sistema los distintos agentes autónomos, entre los que se encuentran el Recuperador, en el cual se integra la herramienta de análisis de videos grabados y el agente autónomo Análisis, el cual ofrece las funcionalidades necesarias para utilizar los video sensores.

Arquitectura en capas:

Arquitectura en capas pertenece a la familia del estilo llamada y retorno, donde cada capa proporciona servicios a la capa superior y se sirve de las prestaciones que le brinda la inferior. Al dividir un sistema en N-capas, cada capa puede tratarse de forma independiente, sin tener que conocer los detalles de las demás. La división de un sistema en capas facilita el diseño modular, la comunicación entre capas está basada en una abstracción que proporciona un bajo acoplamiento entre ellas y con alta cohesión interna, lo cual posibilita variar de una forma sencilla diferentes implementaciones o combinaciones de capas (Reynoso, 2004).

El módulo recuperador implementa una arquitectura en tres capas como una variante del patrón N-capas. Las mismas son denominadas como capa de presentación, capa de negocio y capa de servicio. La capa de negocio, a la cual se integra la herramienta análisis de videos grabados, se encarga de los procesos de recuperación de los videos almacenados.

Arquitectura Modelo-Vista-Controlador:

Según (Bass, 2013) esta es una arquitectura basada en el estilo llamada y retorno, en la cual se separa los módulos en 3 componentes fundamentales, el Modelo, la Vista y el Controlador. Los cuales son tratados como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas. **El Modelo** es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas. **La Vista** es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo. **El Controlador** Contiene el código necesario para responder a las acciones que se solicitan en la aplicación. Sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades del sistema.

Para la implementación de la herramienta de análisis de videos grabados en el Sistema Xilema Suria será utilizada esta arquitectura. Se ha seleccionado debido a que permite obtener una mejor organización en el código de la aplicación, gracias a que esta establece una clara separación entre sus

componentes y facilita su enlace, por lo que facilita la reutilización de los componentes y simplifica el mantenimiento de los mismos. Las clases a implementar se agruparán de la siguiente forma:

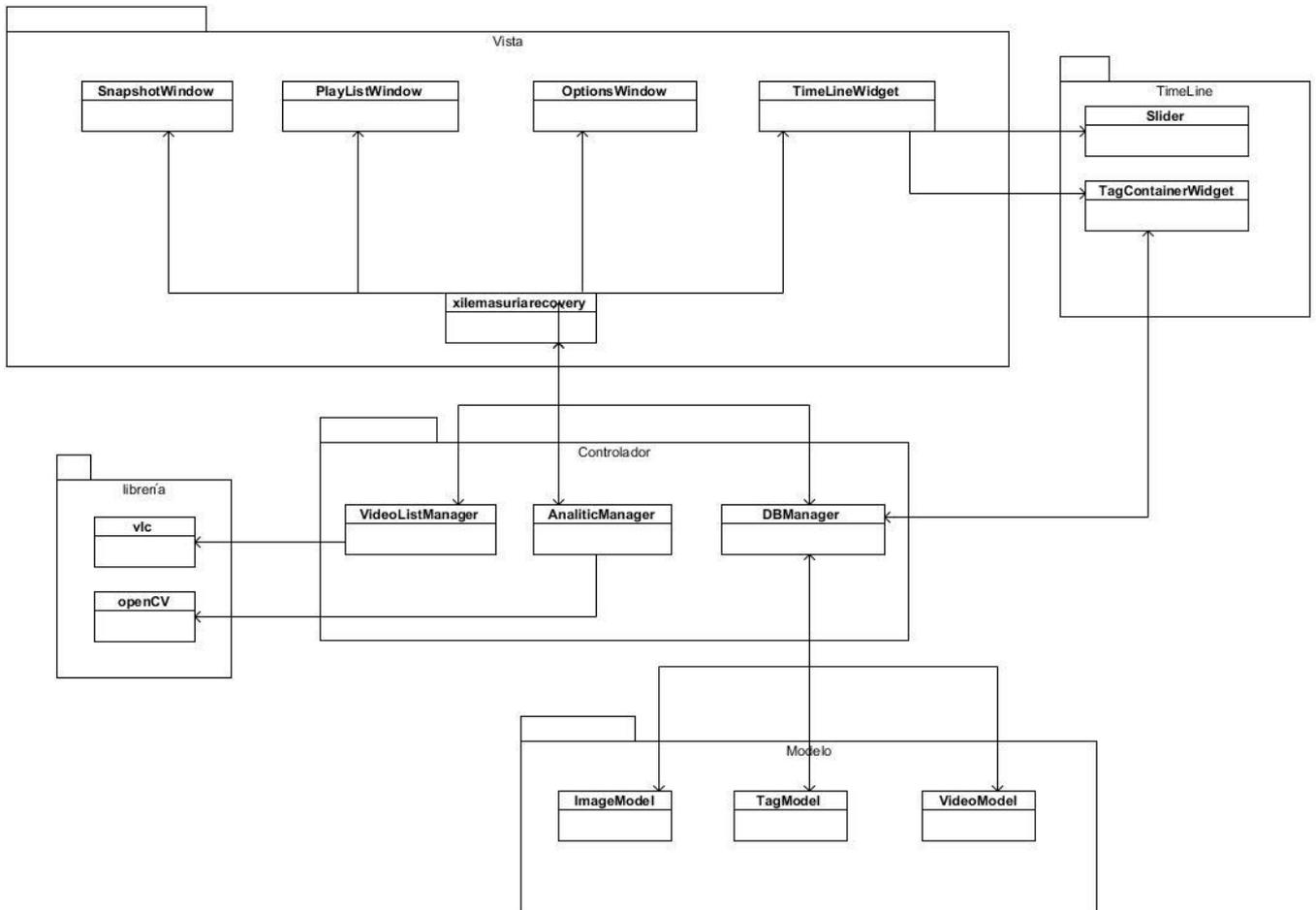


Figura 7. Arquitectura de la herramienta de análisis de videos grabados en el sistema de videovigilancia Xilema Suria.

2.6. Patrones de Diseño:

Los patrones de diseño representan soluciones simples a problemas comunes en el diseño e implementación de software. Estos brindan soluciones probadas a dichos problemas, además facilita entre otras cosas la reutilización y comprensión de códigos. (Gamma, y otros, 2004). Para el diseño de la aplicación se tuvieron en cuenta los Patrones Generales de Software para Asignación de Responsabilidades (GRASP) y los patrones propuestos por *Gang of Four* (GoF).

2.6.1. Patrones GRASP

Patrones GRASP, acrónimo de *General Responsibility Assignment Software Patterns* en español Patrones Generales de Software para Asignación de Responsabilidades, según (Gamma, y otros, 2004) estos describen los principios fundamentales de diseño de objetos para la asignación de

responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Entre los patrones GRASP utilizados están:

- ✓ **Experto:** Este patrón garantiza que cada clase cumpla con sus responsabilidades, y que contenga las funcionalidades que se desean implementar (Larman, 2003). Se evidencia su uso en todas las clases de la aplicación, por ejemplo en la clase *TagContainerWidget* que es la clase experta que conoce todo relacionado con los marcadores.
- ✓ **Bajo Acoplamiento:** Este patrón propone que el desarrollador asigne responsabilidades a clases de tal modo que su colocación no incremente la cantidad de relaciones entre clases (Day, 2013). Este se ve en las relaciones entre las clases *VideoPlayer*, *TimeLineWidget*, *TagContainerWidget*, *Slider*.
- ✓ **Alta Cohesión:** El patrón consiste en valorar cada clase y darle a cada una un alto grado de cohesión, con una importante funcionalidad principal y con poco trabajo que hacer (Day, 2013). Este patrón es usado en todas las clases para asignar responsabilidades.
- ✓ **Creador:** Este patrón guía la asignación de responsabilidades relacionadas a la creación de objetos, por lo que se usa en la creación de una instancia de una clase B en una clase A que la crea o la utiliza (Day, 2013). Se evidencia su uso en la clase *VideoPlayer* en la cual se crean objetos de las clases *PlayListWindow*, *OptionsWindow*, *SnapshotWindow*.
- ✓ **Controlador:** Permite asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas (Day, 2013). Se evidencia su uso en las clases *VideoListManager*, *AnaliticManager* y *DBManager* que son las clases controladoras del sistema.

2.6.2. Patrones GoF

Los patrones GoF (*Gang of Four*, en español Pandilla de los Cuatro), se clasifican en 3 categorías basadas en su propósito: creacionales, estructurales y de comportamiento. Los estructurales tratan la combinación de clases, su relación y la formación de estructuras de alta complejidad, los creacionales tratan la creación de instancias y los de comportamientos tratan la interacción y cooperación entre clases. (Gamma, y otros, 2004) Para el diseño de la herramienta se tuvieron en cuenta los siguientes patrones GoF:

- ✓ **Singleton:** Este patrón asegura que solo exista una instancia de la clase que lo utilice y provee un método para acceder a la misma. Se evidencia su uso en la clase *VideoListManager* que asegura que se devuelva un solo objeto de la misma en todas las clases que la utilicen (Gamma, y otros, 2004).
- ✓ **Observer:** Este patrón define una dependencia del tipo *uno-a-muchos* entre objetos, de manera que cuando uno de los objetos cambia su estado, notifica este cambio a todos los dependientes (Gamma, y otros, 2004). Este patrón es una de las características centrales en Qt, el cual utiliza

Signals y Slot para la comunicación entre objetos, por lo que se evidencia su uso en la aplicación (Qt, 2016).

2.7. Modelo de Diseño

El modelo de diseño describe las interfaces y clases que se van a utilizar. Por lo que sirve como abstracción a la implementación del sistema por lo que facilita el trabajo de desarrolladores (Pressman, 2010).

2.7.1. Diagrama de secuencia del diseño

El diagrama de secuencia que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo, representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino (Jacobson, 2000). Si los CU tienen varios flujos, resulta útil crear un diagrama de secuencia para cada uno de ellos, para una mejor comprensión a la hora de realizarlos (Per Kroll, 2003). A continuación se presentan los diagramas de secuencia de cada uno de los flujos del CU.

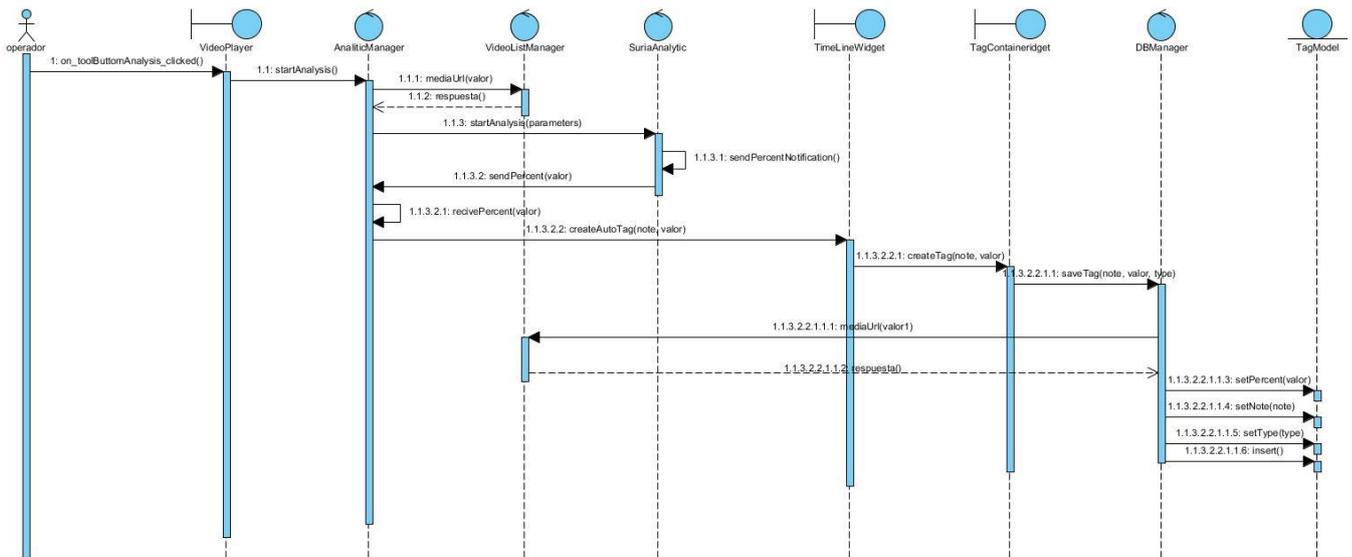


Figura 8. Diagrama de Secuencia de la funcionalidad Iniciar Análisis.

2.7.2. Diagrama de clases del diseño

El diagrama de clases describe la estructura de un sistema ya que muestra clases, métodos, atributos y relaciones que intervienen en cada caso de uso (Pressman, 2010). En la figura 9 se muestra un diagrama de clases del diseño para un caso de uso.

Análisis y Diseño del Sistema:



Figura 9. Diagrama de Clases del caso de uso Realizar análisis.

2.8. Diagrama Entidad Relación:

El diagrama Entidad-Relación es una herramienta para el modelado de datos que permiten representar las entidades que intervienen en el modelo del sistema, así como también sus interrelaciones y propiedades (Jacobson, 2000). A continuación en la figura 10 se presenta el diagrama Entidad-Relación de la herramienta con la descripción de cada una de sus clases.

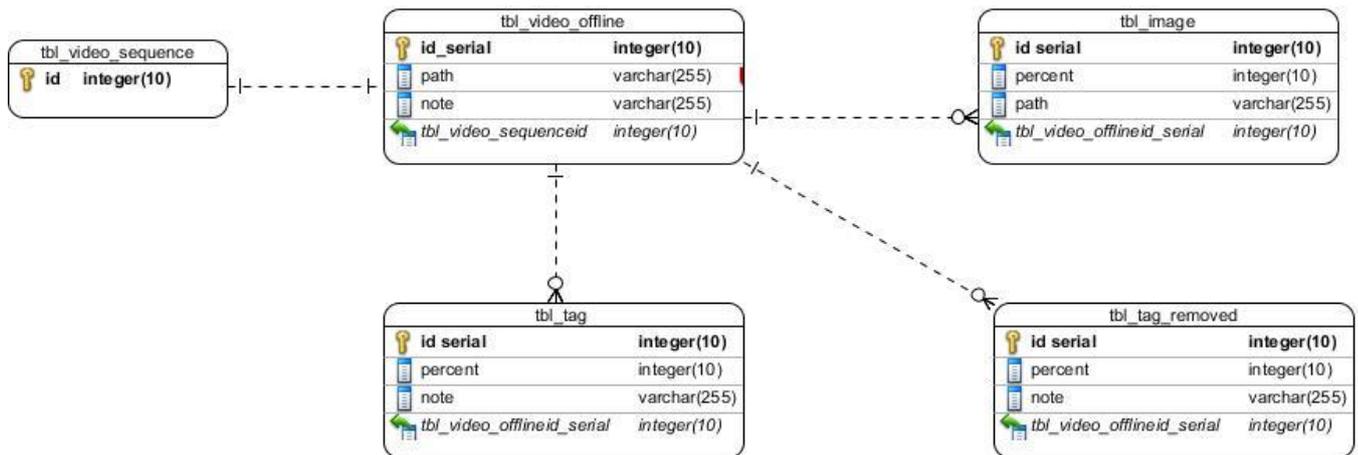


Figura 10. Diagrama Entidad-Relación.

Tabla 3. Descripción de tbl_video_offline.

| | | |
|---------------------|---|--|
| Nombre: | <i>tbl_video_offline</i> | |
| Descripción: | En esta tabla es donde se almacena el id, la dirección y la nota de los videos que se analicen. | |
| Atributos | Tipo | Descripción |
| <i>id_serial</i> | <i>integer(10)</i> | Identificador del video en la Base de Datos. |
| <i>path</i> | <i>varchar(255)</i> | Dirección de donde se encuentra almacenado el video. |
| <i>note</i> | <i>varchar(255)</i> | Nota del video. |

Tabla 4. Descripción de tbl_image.

| | | |
|---------------------|--|---|
| Nombre: | <i>tbl_image</i> | |
| Descripción: | En esta tabla es donde se almacena para cada instantánea el id, la dirección y el porcentaje del instante de captura con respecto a la duración del video. | |
| Atributos | Tipo | Descripción |
| <i>id_serial</i> | <i>integer(10)</i> | Identificador de la imagen en la Base de datos. |

| | | |
|----------------|---------------------|--|
| <i>path</i> | <i>varchar(255)</i> | Dirección de donde se encuentra almacenada la imagen. |
| <i>percent</i> | <i>integer(10)</i> | El porcentaje que representa el tiempo cuando fue capturada la instantánea de la duración total del video. |

Tabla 5. Descripción de *tbl_tag*.

| | | |
|------------------|--|--|
| Nombre: | <i>tbl_tag</i> | |
| Descripción: | En esta tabla es donde se almacena para cada marcador el id, la nota y el porcentaje que representa el tiempo del marcador con respecto al tiempo total del video. | |
| Atributos | Tipo | Descripción |
| <i>id_serial</i> | <i>integer</i> | Identificador del marcador en la Base de datos. |
| <i>percent</i> | <i>varchar(255)</i> | El porcentaje que representa el tiempo donde fue insertado el marcador de la duración total del video. |
| <i>note</i> | <i>varchar(255)</i> | Nota del marcador. |

Tabla 6. Descripción de *tbl_tag_remove*.

| | | |
|------------------|--|--|
| Nombre: | <i>tbl_tag_remove</i> | |
| Descripción: | En esta tabla es donde se almacena para cada marcador eliminado el id, la nota y el porcentaje que representa el tiempo del marcador con respecto al tiempo total del video. | |
| Atributos | Tipo | Descripción |
| <i>id_serial</i> | <i>integer</i> | Identificador del marcador en la Base de datos. |
| <i>percent</i> | <i>varchar(255)</i> | El porcentaje que representa el tiempo donde fue insertado el marcador de la duración total del video. |
| <i>note</i> | <i>varchar(255)</i> | Nota del marcador. |

2.9. Conclusiones parciales:

En este capítulo se especificaron los puntos relacionados con el Análisis y Diseño de la herramienta análisis de videos grabados en el sistema Xilema Suria. Se arribó a las siguientes conclusiones:

Análisis y Diseño del Sistema:

- ✓ La selección de Modelo-Vista-Controlador como arquitectura de la aplicación proporciona una división entre el modelo, la vista y el controlador, que facilita la reutilización del código y el mantenimiento de la herramienta.
- ✓ El uso de patrones de diseño permite desarrollar una solución simple y efectiva para resolver problemas específicos dentro de la herramienta.
- ✓ En el modelo de diseño se logra representar el funcionamiento de la aplicación y las relaciones entre clases mostrando cierto grado de abstracción del código fuente.

Capítulo 3: Implementación y Pruebas:

3.1. Introducción:

En este capítulo se abordan todos los elementos relacionados con el proceso de implementación. Se muestran los diagramas de despliegue y de componentes. Además se describen las pruebas a las que fue sometida la aplicación.

3.2. Modelo de Implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se pueden encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos (Pressman, 2006)

3.2.1. Diagrama de despliegue:

El diagrama de despliegue muestra la disposición física de los distintos nodos¹ que componen un sistema y representan la disposición de los nodos conectados por enlaces de comunicación (Jacobson, 2000). A continuación en la figura 11 se presenta el diagrama de despliegue de la herramienta.

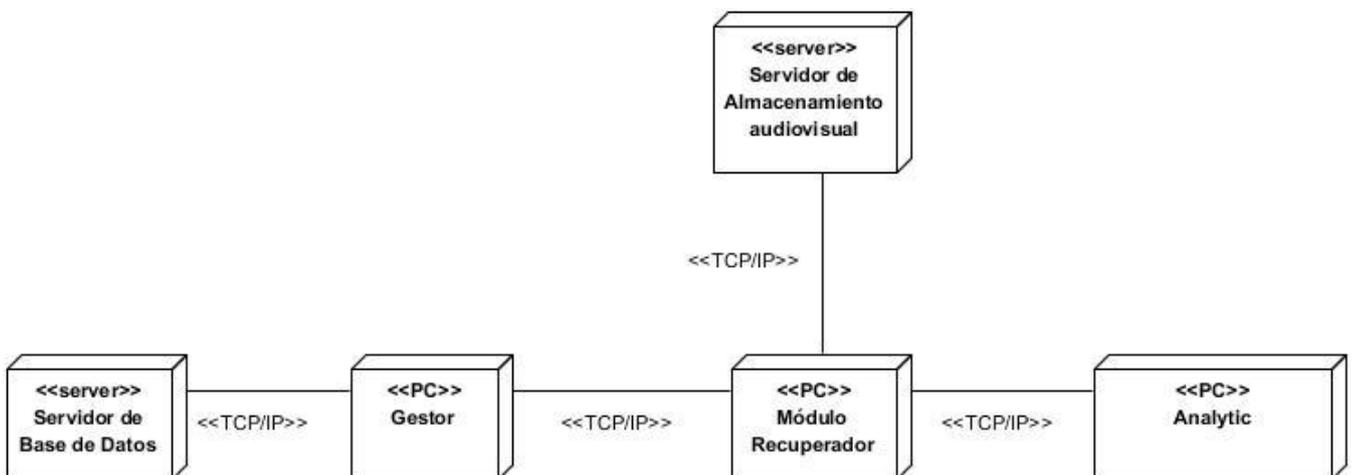


Figura 11. Diagrama de Despliegue

3.2.2. Diagrama de Componentes:

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, sean estos todos los tipos de elementos de software que intervienen en el desarrollo de aplicaciones informáticas. Estos pueden ser simples archivos, código fuentes, paquetes o bibliotecas de

¹ Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria.

desarrollo. En la figura 12 se muestra el diagrama de componentes de la herramienta análisis de videos grabados en el sistema Xilema Suria.

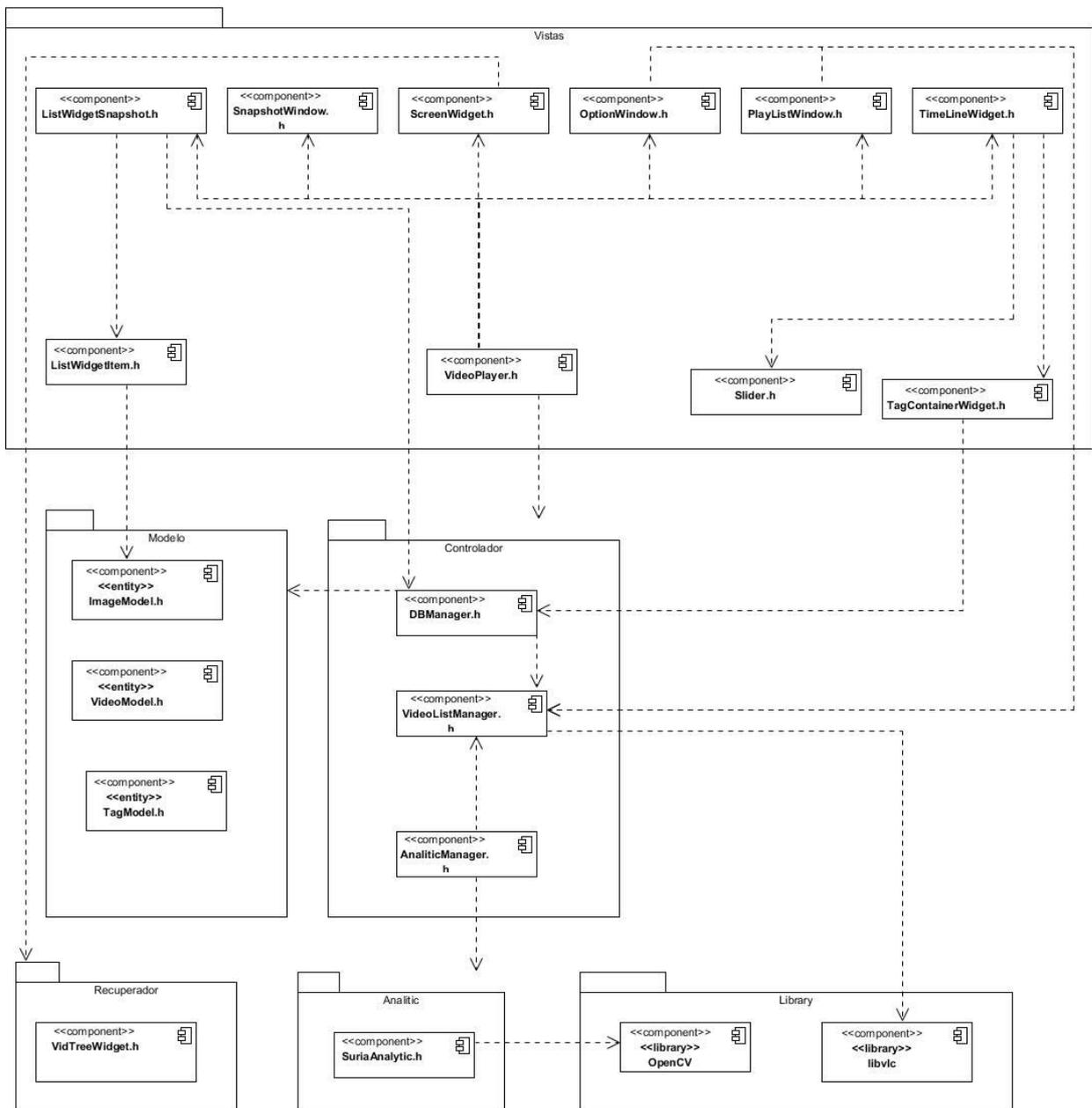


Figura 12. Diagrama de componentes de la Herramienta de análisis de videos grabados en el sistema Xilema Suria.

3.3. Pruebas de Software

Las pruebas de software son un elemento crítico para la garantía de calidad del software y representa una revisión de las especificaciones, del diseño y de la codificación (Pressman, 2010). Para determinar el nivel de calidad se hace necesario realizar diferentes tipos de pruebas que permitan validar las

respuestas del sistema con respecto a cada una de las especificaciones iniciales. En esta etapa se realizan pruebas de Integración y de Sistema.

3.3.1. Pruebas de Integración:

Los módulos de un sistema aun cuando su funcionamiento haya sido probado con anterioridad y respondan bien por separado es necesario probar en conjunto todos los módulos y validar que funcionen correctamente. Dado que al unir estos, pueden no responder de igual manera e incluso pueden llegar a perderse o malinterpretarse datos entre las distintas interfaces que se relacionan (Juristo, 2006).

Para probar que los elementos del software que interactúan entre sí, funcionan de manera correcta se realizan las pruebas de integración con el enfoque incremental ascendente ya que a medida que fueron desarrolladas las funcionalidades se agruparon y se coordinaron las entradas y salidas de los casos de prueba. Se validaron las respuestas dadas tanto por las funcionalidades utilizadas del módulo Análisis como las del Recuperador, de las cuáles el punto crítico fué las respuestas recibidas desde el Recuperador específicamente el árbol que se encarga de mostrar los videos recuperados.

3.3.2. Pruebas del sistema:

Este tipo de pruebas deben ser ejecutadas idealmente por un equipo de pruebas ajeno al equipo de desarrollo. La obligación de este equipo, consiste en la ejecución de actividades de prueba en donde se debe verificar que la funcionalidad total de un sistema fue implementada de acuerdo a los documentos de especificación definidos (Juristo, 2006).

Para la validación de los requisitos funcionales se utiliza el método de Caja Negra. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales del sistema e ignoren la estructura de control del mismo. Existen varias técnicas que permiten realizar pruebas de Caja Negra, para probar la herramienta en cuestión se utilizó la técnica de Partición de Equivalencia esta divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software (Juristo, 2006). A continuación se presentan los casos de prueba para el CU Realizar análisis.

CU4 Realizar análisis.

Descripción general:

Este caso de uso se inicia cuando el operador desea analizar algún video.

Condiciones de ejecución:

El operador debe seleccionar un área de reproducción.

Tabla 1. SC 1 Mostrar sensores disponibles.

| Escenario | Descripción | Respuesta de sistema | Flujo central |
|--|--|--------------------------|---|
| EC1.1 Mostrar sensores. | Selecciona la opción mostrar los sensores disponibles para aplicarle a los videos. | Muestra los sensores. | 1-Herramienta de análisis de video. 2-Selecciona área de reproducción. 3-Selecciona mostrar sensores. |
| EC1.2 Mostrar sensor sin estar seleccionada un área de reproducción. | Selecciona la opción mostrar los sensores. | No muestra los sensores. | 1-Herramienta de análisis de video. 2-Selecciona mostrar sensores. |
| EC1.3 Mostrar sensor sin existir video en el área de reproducción seleccionada. | Selecciona la opción mostrar los sensores. | No muestra los sensores. | 1-Herramienta de análisis de video. 2-Selecciona área de reproducción. 3-Selecciona mostrar sensores. |

Tabla 2. SC 2 Seleccionar sensor a aplicar.

| Escenario | Descripción | Respuesta de sistema | Flujo central |
|--|--|--|--|
| EC1.1 Seleccionar sensor. | Selecciona el sensor con el que se va a analizar el video. | Almacena en una variable el sensor seleccionado. | 1-Herramienta de análisis de video. 2-Selecciona un área de reproducción. 3-Selecciona la opción hacer zoom. |
| EC1.2 Seleccionar sensor sin estar seleccionada un área de reproducción. | Selecciona la opción mostrar los sensores. | No muestra los sensores disponibles. | 1-Herramienta de análisis de video. 2- Selecciona la opción Mostrar sensor. |
| EC1.3 Seleccionar sensor sin existir video en el área de reproducción seleccionada. | Selecciona la opción mostrar los sensores. | No muestra los sensores disponibles. | 1-Herramienta de análisis de video. 2-Selecciona área de reproducción. |

Tabla 3. SC 3 Realizar análisis al video del AR seleccionada aplicando el sensor seleccionado.

| Escenario | Descripción | Respuesta de sistema | Flujo central |
|-----------------------------|---|-----------------------------|--|
| EC1.1 Realizar análisis. | Selecciona el sensor con el que se va a analizar el video y selecciona aceptar. | Realiza el análisis. | 1-Herramienta de análisis de video. 2-Selecciona un área de reproducción. |

Implementación y Pruebas:

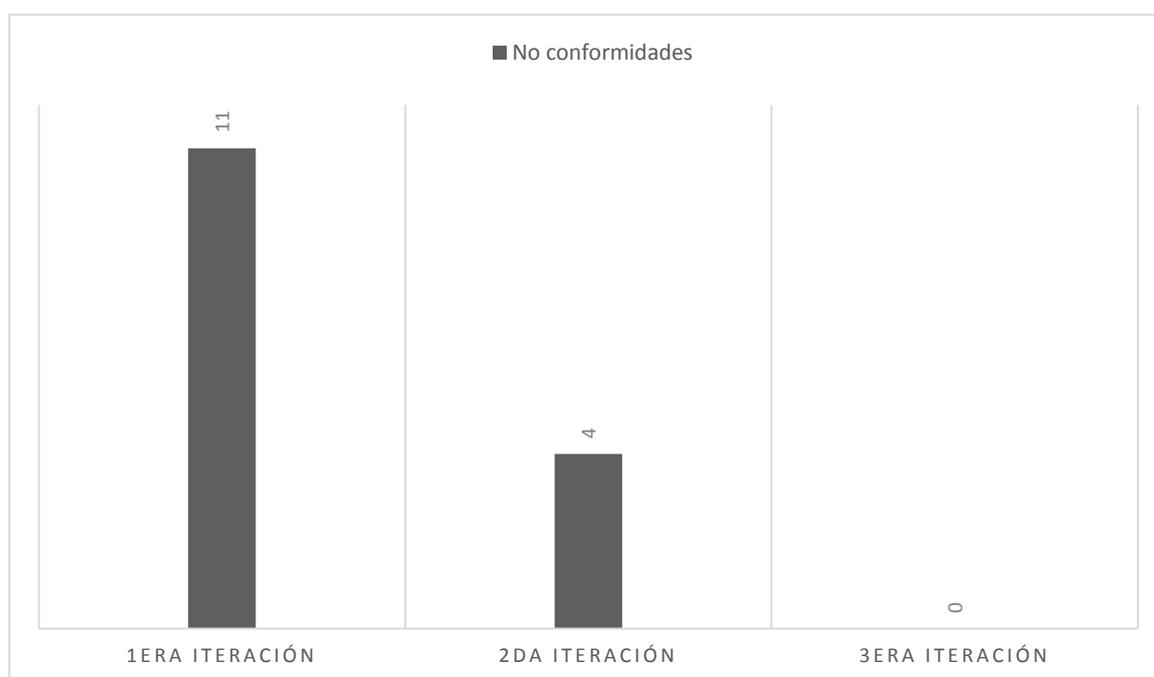
| | | | |
|---|--|--------------------------------------|---|
| | | | 3-Selecciona el sensor. 4- Analiza el video. |
| EC1.2 Realizar análisis sin estar seleccionada un área de reproducción. | Selecciona la opción mostrar los sensores. | No muestra los sensores disponibles. | 1-Herramienta de análisis de video. |
| EC1.3 Realizar análisis sin existir video en el área de reproducción seleccionada. | Selecciona la opción mostrar los sensores. | No muestra los sensores disponibles. | 1-Herramienta de análisis de video. 2-Selecciona área de reproducción. |

Tabla 4. SC 4 Crear marcador por alarmas generadas.

| Escenario | Descripción | Respuesta de sistema | Flujo central |
|--------------------------------|---|--|---|
| EC1.1 Crear marcador. | Selecciona el sensor con el que se va a analizar el video y selecciona aceptar. | Crea un marcador en el instante de tiempo que el video sensor detecte un evento significativo. | 1-Herramienta de análisis de video. 2-Selecciona un área de reproducción. 3-Selecciona el sensor. 4- Analiza el video. 5-Emite una señal. |
| EC1.2 Crear marcador sin estar | Selecciona la opción mostrar los sensores. | No muestra los sensores disponibles. | 1-Herramienta de análisis de video. |

| | | | |
|--|--|--------------------------------------|---|
| seleccionada un área de reproducción. | | | |
| EC1. Crear marcador sin existir video en el área de reproducción seleccionada. | Selecciona la opción mostrar los sensores. | No muestra los sensores disponibles. | 1-Herramienta de análisis de video. 2-Selecciona área de reproducción. |

Figura 13. Resultados de las pruebas del sistema.



En el gráfico antes expuesto se muestran las no conformidades detectadas en las diferentes iteraciones realizadas. La primera iteración arrojó once no conformidades, la segunda cuatro y en la tercera cero. Estas se dividieron entre validaciones y repuestas esperadas desde módulos integrados a la aplicación.

3.4. Conclusiones Parciales:

En este capítulo se especificaron los puntos relacionados con la implementación y prueba de la herramienta análisis de videos grabados en el sistema Xilema Suria. Arribándose a las siguientes conclusiones:

- ✓ Con el desarrollo de los diagramas correspondientes al modelo de implementación, facilitó el entendimiento de la composición del sistema.

Implementación y Pruebas:

- ✓ Al validar la integración de los módulos que se utilizan del sistema Xilema Suria. Además fueron probadas todas las funcionalidades implementadas en la herramienta, obteniéndose once no conformidades en la primera iteración, de las cuales solo cuatro se mantuvieron en la segunda y en la tercera fueron solucionadas todas. Las principales no conformidades fueron detectadas al no estar seleccionado algún video, que la herramienta permitía utilizar los controles del reproductor, los cuáles respondían de forma incorrecta. También fue un punto crítico las respuestas del sistema a validaciones de campos vacíos.

Conclusiones Generales:

Al término de esta investigación, se consideran cumplidos los objetivos y tareas inicialmente propuestos, además se llega a las siguientes conclusiones:

- ✓ El análisis de video inteligente es indispensable para la detección de la ocurrencia de eventos significativos en videos grabados del sistema Xilema Suria.
- ✓ Las herramientas y tecnologías utilizadas en el desarrollo de la aplicación estaban definidas por el proyecto Xilema Suria, lo que favoreció a la integración entre la herramienta y el sistema Xilema Suria.
- ✓ La utilización del patrón arquitectónico Modelo-Vista-Controlador permitió una clara separación entre los componentes del sistema, indispensable para la reutilización de códigos y mantenimiento de la aplicación.
- ✓ Con el desarrollo de la herramienta de análisis de grabaciones del Xilema Suria, se logró aprovechar las funcionalidades de los videos sensores existentes para analizar los videos grabados, además brinda al operador las funcionalidades necesarias para apoyar este tipo de análisis.
- ✓ Con las pruebas realizadas a la aplicación se evidencia que la herramienta cumple con los requisitos propuestos y se integra correctamente con los módulos del sistema Xilema Suria.

Recomendaciones:

Tras haber cumplido con los objetivos propuestos, para mejorar los resultados obtenidos los autores de esta investigación recomiendan:

- ✓ Actualizar los videos sensores existentes o crear nuevos, con el propósito de dotar con nuevas funcionalidades al módulo Análisis del sistema Xilema Suria, para que estas sean explotadas por la herramienta de análisis de videos grabados.
- ✓ Para una mejor experiencia y obtener mejores resultados con la aplicación se recomienda utilizar para su ejecución computadoras con altas prestaciones, especialmente que incluyan tarjetas de video de buen rendimiento.

Referencias Bibliográficas:

- Abdolazimian, Farid y Mansori, Saeedeh. 2008.** *Business Process Reengineering by Rational Unified Process (RUP) Methodology*. s.l. : World Applied Sciences Journal, 2008.
- Alava. 2015.** Alava Seguridad. *ALAVA SEGURIDAD*. [En línea] ALAVA SEGURIDAD, 2015. [Citado el: 21 de enero de 2015.] <http://www.alava-ing.es/seguridad/productos/sistemas-de-videovigilancia--y-cctv/>.
- Albusac, Javier. 2015.** *Aprendizaje de reglas para la clasificación de comportamientos en un sistema de videovigilancia cognitiva*. 2015.
- Aviv, David G. 2013.** *Detection and surveillance system*. U.S : s.n., 2013.
- Bass, Len, Clements, Paul y Kazman, Rick. 2013.** *Software Architecture in Practice 3rd Edition*. New York : Addison Wesley, 2013.
- Bertino, E A y Martino, L A. 1995.** *Sistemas de bases de datos orientadas a objetos*. 1995.
- Carrasco, Jorge. 2010.** *Cine y television digital. Manual Técnico*. Barcelona : Universidad de Barcelona, 2010. 978-84-475-3457-9..
- Conferencia Diseño MIC.** Universidad de las Ciencias Informáticas. 2014. La Habana : s.n., 2014.
- Day, Lewis F. 2013.** *Pattern design*. s.l. : Courier Corporation, 2013.
- G. Medioni, I. Cohen, F. Bremond, S. Hongeng and R. Nevatia. 2001.** *Event detection and analysis from video streams*. s.l. : IEEE, 2001.
- Gamma, Erich, y otros. 2004.** *Design Patterns*. Zúrich : s.n., 2004.
- Gradiant. 2016.** Gradiant. *Gradiant*. [En línea] Gradiant, 21 de febrero de 2016. <http://www.gradiant.org/es/lineas-de-investigacion/procesado-de-senales-multimedia/analisis-de-video-avanzado.html>.
- GsSeguridad. 2016.** GsSeguridad. *GsSeguridad*. [En línea] 3 de abril de 2016. <http://www.gsseguridad.com/video-analisis/>.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000.** *El Proceso Unificado de Desarrollo de Software*. Madrid : Perason Educacion, 2000.
- Josutis, Nicolai M. 2012.** *The C++ standard library: a tutorial and reference*. s.l. : Addison-Wesley, 2012.
- Juristo, Natalia, Moreno, Ana M. y Vegas, Sira. 2006.** *Técnicas de Evaluación de Software*. Madrid : s.n., 2006.
- Larman, Craig. 2003.** *Modelo del Domino*. 2003.

Bibliografías Consultadas:

- Lemus, Pedro Rodríguez López de. 2013.** Blog Canal Profesional. [En línea] 21 de 02 de 2013. <http://pedrorodriguez.blogcanalprofesional.es/videovigilancia/>.
- M. Valera. 2012.** *Intelligent distributed surveillance systems*. 2012.
- OpenCV. 2016.** OpenCV. [En línea] 2016. [Citado el: 21 de febrero de 2016.] <http://docs.opencv.org/2.4.9/>.
- OXFORD University Press. 2016.** oxforddictionaries. *oxforddictionaries*. [En línea] 21 de marzo de 2016. <http://www.oxforddictionaries.com/es>.
- Per Kroll, Philippe Kruchten. 2003.** *The Rational Unified Process Made Easy*. s.l. : Addison-Wesley Professional, 2003.
- pgAdmin. 2016.** pgAdmin. *pgAdmin*. [En línea] 2016. [Citado el: 10 de enero de 2016.] <http://www.pgadmin.org/>.
- PostgreSQL. 2016.** PostgreSQL. [En línea] 2016. <http://www.postgresql.org/docs/9.3/static/index.html>.
- Pressman, Roger S. 2010.** *Ingeniería del Software, Un enfoque Práctico. Sexta Edición*. s.l. : McGraw-Hill., 2010.
- Qt. 2016.** Qt. [En línea] 2016. <https://www.qt.io/>.
- Real Academia Española. 2016.** Real Academia Española. *DRAE*. [En línea] 21 de marzo de 2016. <http://dle.rae.es/index.html>.
- Reynoso, Carlos y Kicillof, Nicolás. 2004.** *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. Buenos Aires : Universidad de Buenos Aires, 2004.
- Rischpater, Ray. 2013.** *Application Development with Qt Creator*. s.l. : Packt Publishing Ltd, 2013.
- Robust Motion Detector for Video Surveillance Applications*.
- Rusu, Lucia, Iuga, Marin y Martis, Simona. 2011.** *Business Process Development Using Agile Methodology*. s.l. : Revista Económica, 2011.
- Sabino, Carlos. 2014.** *El proceso de investigación*. s.l. : Episteme, 2014.
- Sampieri, Hernández. 2010.** *Metodología de la Investigación (Quinta Edición ed.)*. Mexico DF : Mc Graw Hill, 2010.
- Strotrup, Bjarne. 2013.** *The C++ programming language*. s.l. : Pearson Education, 2013.
- VideoLan. 2016.** VideoLan. [En línea] 2016. [Citado el: 23 de enero de 2016.] <https://www.videolan.org/developers/vlc/doc/doxygen/html/index.html>.

Bibliografías Consultadas:

Abdolazimian, Farid y Mansori, Saeedeh. 2008. *Business Process Reengineering by Rational Unified Process (RUP) Methodology*. s.l. : World Applied Sciences Journal, 2008.

Alava. 2015. Alava Seguridad. *ALAVA SEGURIDAD*. [En línea] ALAVA SEGURIDAD, 2015. [Citado el: 21 de enero de 2015.] <http://www.alava-ing.es/seguridad/productos/sistemas-de-videovigilancia--y-cctv/>.

Albusac, Javier. 2015. *Aprendizaje de reglas para la clasificación de comportamientos en un sistema de videovigilancia cognitiva*. 2015.

Álvarez de Zayas, Carlos. 1995. *Metodología de la investigación científica*. Santiago de Cuba : Centro de Estudios de Educación Superior" Manuel F. Gran". Universidad de Oriente, 1995.

Aviv, David G. 2013. *Detection and surveillance system*. U.S : s.n., 2013.

Bándres, Fernando; Delgado, Santiago. 2009. *Biomedicina y derecho sanitario* año. Madrid : Ingraf Impresiones, S.L, 2009. ISBN: 978-84-936577-5-8.

Bass, Len, Clements, Paul y Kazman, Rick. 2013. *Software Architecture in Practice 3rd Edition*. New York : Addison Wesley, 2013.

Bertino, E A y Martino, L A. 1995. *Sistemas de bases de datos orientadas a objetos*. 1995.

Carrasco, Jorge. 2010. *Cine y televidion digital. Manual Técnico*. Barcelona : Universidad de Barcelona, 2010. 978-84-475-3457-9..

Colectivo de Autores. *Conferencia Diseño MIC*. Universidad de las Ciencias Informáticas. 2014. La Habana : s.n., 2014.

Daniel, Rivero. 2013. *Metodología de la investigación*. 2013.

Day, Lewis F. 2013. *Pattern design*. s.l. : Courier Corporation, 2013.

Fernández, Eduardo. 2013. *Security patterns in practice: designing secure architectures using software patterns*. . s.l. : John Wiley & Sons, 2013.

G. Medioni, I. Cohen, F. Bremond, S. Hongeng and R. Nevatia. 2001. *Event detection and analysis from video streams*. s.l. : IEEE, 2001.

Gamma, Erich, y otros. 2004. *Design Patterns*. Zúrich : s.n., 2004.

Gradiant. 2016. Gradiant. *Gradiant*. [En línea] Gradiant, 21 de febrero de 2016. <http://www.gradiant.org/es/lineas-de-investigacion/procesado-de-senales-multimedia/analisis-de-video-avanzado.html>.

Bibliografías Consultadas:

- GsSeguridad. 2016.** GsSeguridad. *GsSeguridad*. [En línea] 3 de abril de 2016. <http://www.gsseguridad.com/video-analisis/>.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000.** *El Proceso Unificado de Desarrollo de Software*. Madrid : Perason Educacion, 2000.
- Jiménez, Ileana Vargas. 2012.** *LA ENTREVISTA EN LA INVESTIGACIÓN CUALITATIVA: NUEVAS TENDENCIAS Y RETOS*. 2012. pág. 21.
- Jordi, Sánchez y Ginés, Benet. 2012.** *Video sensor architecture for surveillance applications*. *Sensors*. 2012. 1509-1528.
- Josutis, Nicolai M. 2012.** *The C++ standard library: a tutorial and reference*. s.l. : Addison-Wesley, 2012.
- Juristo, Natalia, Moreno, Ana M. y Vegas, Sira. 2006.** *Técnicas de Evaluación de Software*. Madrid : s.n., 2006.
- Klette, Reinhard. 2014.** *Concise Computer Vision*. s.l. : Spriger, 2014.
- L. Zelnik-Manor, M. Irani. 2001.** *Event-based analysis of video*. s.l. : Computer Society Conference, 2001. 990935.
- Larman, Craig. 2003.** *Modelo del Domino*. 2003.
- Lemus, Pedro Rodríguez López de. 2013.** Blog Canal Profesional. [En línea] 21 de 02 de 2013. <http://pedrorodriguez.blogcanalprofesional.es/videovigilancia/>.
- León, Rolando Alfredo Hernández y González, Sayda Coello. 2011.** *El proceso de investigación científica*. . s.l. : Editorial Universitaria, 2011.
- Lozano, María Dolores Ruiz. 2010.** *Un modelo para el desarrollo de sistemas de detección de situaciones de riesgo*. Granada : Editorial de la Universidad de Granada, 2010. 9788469411803.
- Lyon, David y Ball, Kirstie. 2012.** *Routledge handbook of surveillance studies*. s.l. : Routledge, 2012.
- M. Valera. 2012.** *Intelligent distributed surveillance systems*. 2012.
- Malmenbrat, Johan y Brooks, David J. 2015.** *CCTV surveillance: The differing aims and functions of CCTV*. 2015.
- Meurant, Gerard. 2013.** *CCTV Surveillance: Video Practices and Technology*. s.l. : Elsevier, 2013.
- OpenCV. 2016.** OpenCV. [En línea] 2016. [Citado el: 21 de febrero de 2016.] <http://docs.opencv.org/2.4.9/>.
- OXFORD University Press. 2016.** oxforddictionaries. *oxforddictionaries*. [En línea] 21 de marzo de 2016. <http://www.oxforddictionaries.com/es>.

Bibliografías Consultadas:

- Per Kroll, Philippe Kruchten. 2003.** *The Rational Unified Process Made Easy*. s.l. : Addison-Wesley Professional, 2003.
- pgAdmin. 2016.** pgAdmin. *pgAdmin*. [En línea] 2016. [Citado el: 10 de enero de 2016.] <http://www.pgadmin.org/>.
- PostgreSQL. 2016.** PostgreSQL. [En línea] 2016. <http://www.postgresql.org/docs/9.3/static/index.html>.
- Pressman, Roger S. 2010.** *Ingeniería del Software, Un enfoque Práctico. Sexta Edición*. s.l. : McGraw-Hill., 2010.
- Prezi. 2016.** Prezi. *Prezi*. [En línea] 2016. [Citado el: 14 de mayo de 2016.] <https://prezi.com/xkwi6hgfjog/funciones-de/>.
- Qt. 2016.** Qt. [En línea] 2016. <https://www.qt.io/>.
- Real Academia Española. 2016.** Real Academia Española. *DRAE*. [En línea] 21 de marzo de 2016. <http://dle.rae.es/index.html>.
- Reynoso, Carlos y Kicillof, Nicolás. 2004.** *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. Buenos Aires : Universidad de Buenos Aires, 2004.
- Rischpater, Ray. 2013.** *Application Development with Qt Creator*. s.l. : Packt Publishing Ltd, 2013.
- Robust Motion Detector for Video Surveillance Applications*. **A. Albiol, C. Sandoval. 2003.** 2003, Proc. of the International Conference on image processing.
- Rusu, Lucia, Iuga, Marin y Martis, Simona. 2011.** *Business Process Development Using Agile Methodology*. s.l. : Revista Económica, 2011.
- Sabino, Carlos. 2014.** *El proceso de investigación*. s.l. : Episteme, 2014.
- Saini, Mukesh K. 2013.** *Privacy aware publication of surveillance video*. . s.l. : International Journal of Trust Management in Computing and Communications, 2013.
- Sampieri, Hernández. 2010.** *Metodología de la Investigación (Quinta Edición ed.)*. Mexico DF : Mc Graw Hill, 2010.
- Sanchez, Tamara Rodriguez. 2015.** *Metodología de desarrollo para la actividad productiva en la UCI*. La Habana : UCI, 2015.
- Scribd. 2016.** Scribd. *Scribd*. [En línea] Scribd, 2016. <http://es.scribd.com/doc/27239031/PATRONES-DE-DISEÑO>.
- Strotrup, Bjarne. 2013.** *The C++ programming language*. s.l. : Pearson Education, 2013.

Bibliografías Consultadas:

Torres, Carlos M. Falcón. 2012. *Sistema de Vigilancia Autónomo Multiagente*. Las Palmas de Gran Canaria : Universidad de las Palmas de Gran Canaria, 2012.

VideoLan. 2016. VideoLan. [En línea] 2016. [Citado el: 23 de enero de 2016.] <https://www.videolan.org/developers/vlc/doc/doxygen/html/index.html>.

Visual Tools. 2016. Visual Tools. *Visual Tools*. [En línea] 4 de marzo de 2016. <http://www.visual-tools.com/productos/analisis-de-video-conteo-de-personas-y-control-pos>.

Zhang, Cheng y Budgen, David. 2013. *A survey of experienced user perceptions about software design patterns*. s.l. : Information and Software Technology, , 2013. 822-835.

