UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 6

Título: Plugin generador de StoryBoards para el Sistema Gestor de Procesos de Media

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

Autor

Elisa Lisbeth Vázquez Chacón

Tutor

Ing. Janet Cristina Labrada Paneque

Ciudad de la Habana, junio de 2013

"Año de 55 de la Revolución"

"La perfección de los medios y la confusión de los fines parecen caracterizar a nuestra época."

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor del trabajo **"Plugin generador de StoryBoards para el Sistema Gestor de Procesos de Media"** y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

_____ _____

Elisa Lisbet Vázquez Chacón Ing. Janet Cristina Labrada Paneque

Autor **Tutor**

DATOS DE CONTACTO

Tutor: Ing. Janet Cristina Labrada Paneque

Formación académica:

Ingeniero en Ciencias Informáticas, Universidad de Ciencias Informáticas, 2012.

Centro Laboral:

Centro de Desarrollo de Geoinformática y Señales Digitales. Facultad 6.

Correo electrónico: janetcristina@uci.cu

Agradecimientos

@ mi mama Arianne, por ser mi padre, mi amiga, mi hermana. Por siempre estar ahí en cada tropiezo, en cada caída, y siempre ayudar a levantarme, gracias por secar mis lágrimas en cada momento que te necesité.

@ mi abuela Elba por siempre pelearme y tratar de llevarme por el camino correcto, gracias a ella hoy soy quien soy.

@ mis abuelos Cristino y Dora, por ser mis padres, y por siempre cuidar de mí.

@ mi tía Yodalis, que diosito me la tenga en la gloria, por ser todo para mí.

@ mis hermanos Leo,Jesusito, Sandra y mis primos Yissel, Kevin, Kayla y Enricel por siempre soportarme.

@ los Jesús por siempre estar ahí cada vez que los necesitaba.

@ mi familia en general por siempre guiarme por el camino correcto y por apoyarme durándote toda la carrera a pesar que estuve lejos de ellos por estos 5 años.

@ mi tutora Janet (Juana), por siempre confiar en mí, y por ayudarme a secar las lágrimas cada vez que salía de los corte de tesis.

@ todos los profesores de mi proyecto, Dainovys, Jean Michel, y a Adnan Fuentes por tener tanta paciencia conmigo, y explicarme las cosas varias veces hasta que las entendiera, gracias.

@ mi oponente Pupo, por siempre ayudarme.

@ mis amigas Dolores, Dairelis y Saydi, por ser siempre mis mejores amigas, y por estar ahí cada vez que las necesitaba. Dolores en especial a ti por ayudarme a no irme de la escuela en 2do año cuando así lo tenía pensado, y por siempre brindarme tu casita para quedarme.

@ todas las chicas y chicos del 93104 por hacerme sentir que estaba en mi casa, y no permitirme extrañar la misma, me llevo buenos recuerdos de todos.

A Joel por siempre estar ahí cada vez que lo necesitaba, gracias.

A todos mis amigos de la fac. 1 por siempre estar ahí en cada fiesta, también me llevo súper recuerdos de todos.

A todos mis compañeros de aula, por siempre ayudarme cada vez que los necesitaba, especialmente, a Yuniel, Adrián, Javier, Orisel, Dany, Doina.

A todos mis amigos en general, por hacerme vivir los mejores momentos de mi vida, acá en la escuela.

A todos los profesores que compartieron conmigo estos 5 años, y contribuyeron en mi educación para formarme como profesional.

Me voy feliz, y agradecida con la vida, el haberme permitido conocer todas las personas que hoy realmente conocí aquí, y espero que la amistad no quede solo acá, aunque no seamos de los mismos lugares y quizás más nunca en la vida tropecemos de nuevo, pero quiero que todos me recuerden de la mejor manera.

Dedicatoria

Dedico esta tesis a todas estas personas lindas que confiaron en mí.

A mi madre y a mis hermanos, por su dedicación y por todo el amor que siempre me han brindado.

A mis abuelos, mis primos por siempre estar ahí conmigo.

A mi familia por todo el apoyo que me han entregado.

A todos mis amigos por permitirme compartir tiempo con ellos.

A mí, por todo el esfuerzo de estos 5 años.

Resumen

El análisis de grandes volúmenes de video requiere de tecnologías potentes para el procesamiento de cada una de las partes que lo componen. Por cada segundo se procesan decenas de imágenes, por lo que resulta poco factible manipular todos estos datos de manera simultánea, si se tiene en cuenta la complejidad de los algoritmos de procesamiento existentes. El presente trabajo de diploma contiene la investigación y el proceso de desarrollo realizado para obtener un componente de software que permita, integrado al Sistema Gestor de Procesos de Media (SGPM), separar los fotogramas claves de los videos para que pueda ser utilizado por todos los demás proyectos con los que se cuenta en el departamento. Se propone un método basado en umbrales para el proceso de detección de cambio de tomas. Se realiza un agrupamiento jerárquico para obtener como resultado el resumen de video. Con el fin de obtener los mejores resultados el desarrollo se realiza bajo la guía de la metodología *OpenUP*, la aplicación es implementada haciendo uso del lenguaje de programación C++, como *framework* de desarrollo QT y como biblioteca de apoyo para el procesado de los fotogramas del video se utiliza *OpenCV*. Se presentan los requisitos identificados y su cumplimiento en la aplicación final, que son validados además por las pruebas realizadas. El impacto social de los resultados de esta investigación se centra en un componente que permita separar de manera automática segmentos de videos en forma de *storyboard* para solucionar los problemas presentes en el departamento Señales Digitales.

Palabras Claves:

Agrupamiento, componente, fotograma clave, *storyboard*, toma, umbrales, video

Índice

Índice de figuras

Índice de tablas

Introducción

A la luz de los avances tecnológicos en la producción de archivos de imágenes en movimiento, (video), y frente al uso de los mismos, cada día se afianza la tendencia que acepta que un video no es solo una reproducción fiel a la realidad, sino que se le reconoce como la representación y reconstrucción de una secuencia de imágenes, que pueden incluir sonido. El video no es solamente una manera de observar, estudiar o analizar el mundo a través de imágenes y sonidos, sino que además es también la tecnología capaz de captar y procesar una serie de fotografías que luego muestran una secuencia, y a una gran velocidad reconstruyen la escena original (González, 2011).

Dentro del marco tecnológico del video, la digitalización ha irrumpido con gran fuerza. Lograr el análisis de grandes volúmenes de videos resulta un problema complejo, pues se requiere de tecnologías avanzadas para el procesamiento de cada una de las partes que componen el video. Por tales razones, la tecnología de búsqueda en video se está convirtiendo rápidamente en una actividad necesaria para el estilo de vida actual, pues más consumidores tienen acceso a conexiones internet de alta velocidad y consumen más contenido de audio y video que antes.

La generación de archivos en formatos de videos ha sido llevada a las aplicaciones que se dedican a transmitir informaciones a través de la red, combinando así elementos que pueden visualizarse con mayor claridad. Existen enormes esfuerzos por desarrollar algoritmos de segmentación de videos, los cuales son bases de cualquiera de las técnicas existentes para el análisis de información contenida dentro de diferentes formatos de videos. Estos algoritmos deben generar resúmenes, en videos de grandes tamaño, de modo que queden reflejados los elementos de mayor interés como parte del mismo.

La Universidad de las Ciencias Informáticas (UCI), fue creada en el 2002 con el objetivo de formar profesionales comprometidos con la Patria cubana, en función del desarrollo de la producción de software. El centro de desarrollo Geoinformática y Señales Digitales (GEySED), que se encuentra en la facultad 6 de la UCI, cuenta con el departamento Señales Digitales creado con el fin de la gestión de los archivos audiovisuales, donde la utilización de estos, provoca la necesidad de poseer gran cantidad de funciones que permitan la realización de una manipulación profunda de los archivos almacenados en los diferentes proyectos del departamento.

La representación de los elementos predominantes de un video es una de las cuestiones que se trabaja en el departamento para poder realizar resumen de video. El resumen de video representa un beneficio notable para los usuarios del sistema ya que no tendrán que visualizar por completo un material para

conocer el contenido del mismo. En el futuro se desea que los usuarios que consuman de estos servicios audiovisuales en cualquier canal, plataforma o sistema televisivo tengan la posibilidad de seleccionar el material que desean adquirir o reproducir sin tener que visualizarlo completamente, sino solamente mostrando sus elementos fundamentales que evidencia de manera clara el contenido que posee.

Uno de los sistemas que son desarrollados en el departamento es el SGPM. Es un producto que está enfocado a la integración con estructuras de software y hardware que existan en el lugar donde se implemente. Garantiza interoperabilidad entre sistemas coexistentes y optimiza los tiempos de integración entre sistemas de diversas índoles. Está compuesto por 4 subsistemas, editor de flujo de procesos, gestor de flujos de procesos, plataforma de codificación e indexación y monitor de procesos de media; estos se encargan de lograr la integración con disímiles sistemas y hardware especializados para la producción audiovisual, además de que mantienen una arquitectura desacoplada que permite la puesta en marcha del sistema con requisitos altamente variables.

Como el SGPM se encarga de gestionar las operaciones automáticas que provienen de otros sistemas de descripción audiovisual, se requiere que la representación de video forme parte de las funciones del SGPM para que su utilización sea posible por todos los sistemas desarrollados en el departamento, y así ampliar su escalabilidad y alcance a otros proyectos que necesiten de esta funcionalidad.

Partiendo de lo anteriormente descrito se plantea como **problema a resolver:** ¿Cómo garantizar la representación de contenidos de video en forma de *Storyboard* para el SGPM? Y para ello se ha definido como **objeto de estudio,** los componentes para la representación de resúmenes de video. Como **campo de acción** se define el componente para la representación de resúmenes de video en forma de *storyboard* para el SGPM.

El **objetivo general** de esta investigación es implementar un componente de software que permita separar de manera automática segmentos de video para su representación en forma de *storyboard*, integrado al SGPM.

Para ello se plantea como **idea a defender** que si se implementa un componente para la separación automática de video integrado al SGPM se garantizará resumir la información visual en forma de *storyboard*.

Para dar respuesta al objetivo propuesto se trazaron las tareas que a continuación se mencionan:

- Elaboración del marco teórico sobre la realización de resúmenes visuales de video.
- Definición de las herramientas y tecnologías a utilizar.

- Modelación de la solución propuesta para la realización de resúmenes visuales de video.
- Implementaciónde la solución parala realización de resúmenes visuales de video.
- Realización de pruebas de integración y de rendimiento al componente desarrollado.

Para dar cumplimiento a las tareas planteadas anteriormente se utilizaron diferentes **métodos científicos** durante el desarrollo de la investigación.

Entre los **métodos teóricos** a utilizar están:

Analítico - Sintético: Permite llegar a conclusiones a partir de las diferentes fuentes bibliográficas utilizadas, para poder definir una propuesta para el sistema de resumen de video propuesto.

Análisis histórico lógico: Permite conocer el comportamiento y evolución de los diferentes enfoques existentes en la separación automática de video, así como los métodos propuesto para su evaluación.

Modelación: Se emplea para crear una abstracción del mundo real y modelarlo de la forma más conveniente, partiendo de la situación problemática de la investigación. Esto se logrará con el lenguaje de modelado UML que permiten representar y manejar los elementos reales y abstraerlos hasta el punto de poder manejarlos de una forma más conveniente para su implementación.

El **método empírico** utilizado fue:

Experimental: Permite la validación de los resultados obtenidos durante la investigación a partir de la selección de un conjunto de métricas que permitan evaluar la eficiencia del algoritmo.

Estructura del documento:

En el Capítulo 1 se hace una valoración de los diferentes enfoques existentes relacionados con segmentación automática de video. Se describen los conceptos asociados a la separación automática de video, además de todas las herramientas y lenguajes necesarios para la investigación. También se hace un análisis profundo de varias soluciones existentes relacionadas con el tema con el objetivo de obtener mejoras que pueden ser utilizadas para dicho componente.

En el Capítulo 2 se hace la descripción del algoritmo, se definen los requisitos funcionales y no funcionales que debe tener el futuro componente, así como la realización de sus diagramas de clase, de secuencia y colaboración para su posterior implementación.

En el Capítulo 3 se hace la implementación del componente. Además de realizar las pruebas de integración y de rendimiento, donde se demuestra que la solución está funcionando correctamente.

Capítulo 1: Fundamentación teórica

En este capítulo se analiza el proceso de resumen de video, sus principales características y conceptos asociados. Además de explicar los aspectos relacionados a la situación problemática presentada, así como se analizarán las posibles soluciones existentes y las tecnologías y herramientas que guían el desarrollo de la investigación.

1.2 Conceptos asociados al dominio del problema

En el proceso de realizar resumen de video de archivos se relacionan todo un grupo de conceptos asociados al mismo. A continuación se muestra una serie de términos que ayudarán a una mejor comprensión del contenido de la investigación.

Plugin

Se hace necesario para esta investigación utilizar este tipo de componente debido a que, un *plugin* es una aplicación informática que añade funcionalidades específicas a un programa principal. Su presencia es muy habitual en los navegadores web, en reproductores de música y en sistemas de gestión de contenidos. En el caso de este *plugin* que se ha de desarrollar, será utilizado por todos los proyectos pertenecientes al departamento que así lo requieran (KIOSKEA, 2012).

Video digital

El video digital consiste en mostrar una sucesión de imágenes digitales. Dado que estas imágenes digitales se muestran a una frecuencia determinada, es posible saber el número de bytes mostrados (o transferidos) por unidad de tiempo. De esta manera, la frecuencia necesaria para mostrar un video (en bytes por segundo) equivale al tamaño de la imagen multiplicado por el número de imágenes por segundo. Sea V una secuencia de fotogramas $I1\,I2\ldots In$ que son indexados a una frecuencia de grabación constante fps dando una sensación de movimiento a los contenidos que se encuentran en cada imagen. En la definición Ii representa el fotograma $i-th$ dentro del video, n indica la cantidad de fotogramas del video, mientras fps la cantidad de fotogramas que componen el video por cada segundo. Cada fotograma representa la unidad básica de información del video (Wah, 2005).

Resúmenes de video

Las técnicas de resúmenes automáticos de video tienen como propósito obtener, a partir de un video, una versión reducida del mismo con los contenidos más relevantes. Existen dos modalidades de resúmenes de

video que son las que mayor aceptación han tenido por los usuarios de estos sistemas. Estos tipos de resumen de video son los *Storyboards* y los *Skimming* de video (Wandelmer, 2006).

Storyboard

El *storyboard* es una herramienta útil para la elaboración de guiones, tanto del género dramático como el género informativo. Consiste en una serie de pequeños dibujos ordenados en secuencia de las acciones que se van a filmar o grabar, de manera que la acción de cada escena se presenta en términos visuales. Ayuda a visualizar las ideas del guionista y es muy utilizado en la producción de anuncios comerciales, videoclips, audiovisuales de transparencias y películas con diseños visuales muy elaborados. Aunque parezca extraño, es una herramienta muy útil en la elaboración de historias dramáticas y anuncios comerciales para radio. Ayuda a que los elementos aurales se visualicen y se combinen de una mejor manera. En el *storyboard*, cada dibujo va acompañado de un comentario descriptivo de la acción, narración o diálogo. El producto final es muy parecido a una tira cómica, con viñetas individuales que presentan las imágenes importantes del desarrollo de la historia (Medrano, 2009).

Skimming de video

Consisten en una agrupación de pequeños segmentos de videos, donde cada segmento contiene la información más relevante del video. Estos pequeños segmentos de video se conectan a partir de diferentes tipos de efectos visuales para obtener un resumen del video original.

Para obtener el resumen de video en algunas de las modalidades anteriormente expuestas es necesario tener en cuenta la estructura del video por lo que existen tres pasos fundamentales para obtener el resumen de video (Saez, 2006).

- Extracción de los fotogramas claves a partir de la detección de los cambios de tomas.
- El análisis de la información redundante en el video.
- La estrategia para la generación de los resúmenes de video.

La presente investigación se centrara en los resúmenes de videos storyboards, debido a que este tipo de resumen es el que satisface las necesidades del SGPM.

Toma de un video

Se define como toma de video a la unidad sintáctica mínima en la que se estructuran los videos. Comprende las imágenes registradas durante una operación de la cámara, esto es, desde que se pulsa el

botón de grabación hasta que se detiene la acción (Wandelmer, 2006). Una misma toma puede contener varios planos.

Cambios de tomas

La transición entre dos tomas, es decir la unión de dos de estas, aparece en el proceso de post-producción mediante técnicas de edición de video. En tal sentido el conjunto de tomas que fueron capturadas por una cámara se editan para eliminar partes de la toma que resultan contenido indeseado (errores de grabación, etiquetas usadas en la grabación, imágenes que quedaron en negro o con el patrón de la cámara, etc.). Por último se unen estas tomas, para conformar el video final, utilizando efectos de edición los cuales son conocidos como transiciones. Esta transición puede será abrupta o gradual dependiendo del efecto que se quiera lograr. En el caso de las transiciones abruptas es el resultado de un efecto de corte en el video. En este sentido se dice que existe una transición abrupta si los fotogramas lk e $lk + 1$pertenecen a tomas diferentes. Esta transición se detecta claramente debido a la discontinuidad visible en la secuencia de fotogramas. Este tipo de transiciones son conocidas por corte. En el caso de las transiciones graduales es el resultado de la aplicación de efectos de edición de video más complejos, en los que se ven involucrados varios fotogramas. Por lo tanto se dice que existe una transición gradual si los fotogramasIk e $Ik + N$ pertenecen a tomas diferente. Donde N representa el número de fotogramas que conforman la transición. Algunos ejemplos de estas transiciones son los *fade*, *dissolves*.

Cambios de tomas abruptos

Se caracterizan porque emplean un solo fotograma para enlazar dos tomas. Son las transiciones más abundantes existentes en los videos y existe únicamente un tipo de efecto de edición representativo de las transiciones abruptas, el corte (Saez, 2006). La detección de este tipo de cortes es un problema ampliamente estudiado y muy fácil para hacer frente en la mayoría de los casos. El porcentaje de éxito reportado por los autores se encuentra actualmente por encima del 95% de *recall* y 95% de precisión, calculado sobre una muestra máxima de alrededor de 300 cortes. Sin embargo, no es fácil comparar los resultados reportados objetivamente (Bescòs Cisneros, y otros, 2006).

Cambios de tomas graduales

Estos cambios de tomas no se realizan usando un solo fotograma como lo hacen los abruptos, sino que, utilizan varios fotogramas para cambiar de una toma a otra. Esta unión gradual entre tomas se realiza con la aplicación de varios efectos de edición como pueden ser el *dissolve*, que es posiblemente el más utilizado, los *fades (fade-in y fade-out), wipes, matte* (Saez, 2006). Con frecuencia son utilizados para

incorporar información semántica al video, como por ejemplo para indicar un cambio de escena. Su detección es sustancialmente más complicada que la detección de transiciones abruptas, pues al ocurrir de forma gradual quedan habitualmente enmascaradas por el movimiento presente en el video (Lucas HB, y otros, 1999). Además, la gran variedad de transiciones graduales existentes es otro aspecto que dificulta su detección. Por esto la gran complejidad de los algoritmos que se usan para detectarlos.

Etapas para la detección de cambios de tomas

Los algoritmos de detección de cambios de tomas se basan en el cálculo de la diferencia entre las características visuales de los fotogramas Ik y$Ik + N.$

En el caso particular en que $N = 1$ los fotogramas son consecutivos. Existen tres enfoques fundamentales para detectar cambios de tomas que durante varios años (2000-2007) han sido objeto de prueba en los eventos TRECVid[1].

- Análisis de Histograma (15 grupos han utilizado este enfoque).
- Análisis de los valores de intensidad del píxel (8 grupos han utilizado este enfoque).
- Análisis de bordes (5 grupos han utilizado este enfoque).

La forma en que se han empleado cada una de estas métricas van desde métodos donde se utilizan cada una de ellas de manera independiente hasta métodos más complejos donde se combinan varias de ellas para obtener una métrica más robusta. De manera particular cuatro de los grupos con mejores resultados en el TRECVid 2005 utilizaron la combinación de estas tres técnicas como parte de sus algoritmos de detección de cambios de tomas (Smeaton, y otros, 2010). Como salida de estos, se obtiene un vector de distancias por cada métrica empleada con valores cercanos para aquel par de imágenes que pertenecen a una misma toma y ocurre un cambio brusco cuando aparece una nueva toma.

Luego de la extracción de características es necesario desarrollar una estrategia de clasificación que permita separar este vector de características en dos clases, transición y no transición.

1.3 Análisis del objeto de estudio

Para el desarrollo de esta investigación se definió como objeto de estudio, las representaciones de videos en forma de resumen.

[1] Conferencia de pruebas de recuperación. El objetivo de este ciclo de conferencias es fomentar la investigación en recuperación de información, proporcionando una gran colección de pruebas, procedimientos uniformes de puntuación, y un foro para las organizaciones interesadas en la comparación de sus resultados con años anteriores

Descripción general

El video es la tecnología capaz de captar, procesar y analizar una secuencia de imágenes, para el procesamiento del mismo se hace necesario poseer herramientas avanzadas, para lograr el análisis de cada una de las partes que conforman al mismo. Frente al uso de estos, se han incrementado el número de operaciones que se pueden realizar sobre ellos, como pueden ser, sobrescribir, codificar, indexar, resumir y así poder sacar información de cada uno de los fotogramas con lo componen. El proceso resumir video se realiza sobre los fotogramas del mismo, mostrando solo aquellos que son claves, y poseen la mayor cantidad de información de video.

Estos fotogramas del video es lo mismo a lo que hoy se conoce como Storyboard, por lo que se tienen todas las imágenes con las que cuenta el video. Existen varias compañías que realizan los storyboard, con el objetivo de poder obtener una secuencia de lo que se desea grabar. La mayoría de estas compañías son privativas por lo que se desconoce la forma en que las mismas realizan este proceso de extracción de fotogramas.

A continuación se muestran algunas estrategias de resumen de video que fueron útiles para esta investigación.

Estrategias de resúmenes de video

Las estrategias de resumen de video han sido abordadas desde diversos enfoques. Algunos autores trabajan los métodos de resumen de video sobre la base de obtener como salida un único resumen del video. Sin embargo trabajos más actuales se dirigen hacia la generación jerárquica de varios resúmenes para un mismo video de modo que se explote al máximo las estructuras del mismo (Benini, 2006). Sobre esta base (Jimenez, 2004) introduce el concepto de escalabilidad con el propósito de obtener múltiples resúmenes de video a partir de un agrupamiento jerárquico de los fotogramas claves. Este concepto resulta de mucha importancia en la navegación por los fotogramas del video y dependiendo del nivel de detalle que se desee por el usuario se obtendrá mayor o menor información.

Existen disímiles (Benini, 2006) (Chen, 2010) trabajos que utilizan el agrupamiento jerárquico de fotogramas para escalar los resúmenes de video. La idea general de estos trabajos consiste en construir una jerarquía de fotogramas siguiendo el criterio del vecino más cercano. Se utilizan medidas visuales de distancias similares a las descritas en la sección de cambios de tomas para construir la matriz de distancia

entre fotogramas. Con el dendograma[2] construido se aplica algún criterio de umbral para definir las agrupaciones que se generan. A partir de estructurar las tomas en grupos se siguen diversas estrategias para escalar los niveles de resumen de los videos.

En el trabajo se utiliza a partir de la construcción de los grupos de tomas una medida estadística, con pesos por cada componente, sobre la base de la duración de cada grupo y la distancia entre los fotogramas que componen el grupo. Esta medida estadística es un índice de relevancia de cada grupo el cual se utiliza para las diferentes salidas que el usuario requiere como parte del resumen del video. De manera similar se presenta una medida de relevancia (Chen, 2010) con la diferencia que el autor trabaja de manera directa sobre las tomas detectadas.

(Benini, 2006) utiliza como criterio de distancia para el agrupamiento jerárquico de cada fotograma clave extraído, un *tree search vector quantizer* TSVQ[3] *codebook* el cual contiene información semántica de cada fotograma clave seleccionado. Con esto se construye una medida de dispersión la cual se emplea tanto para construir la distancia entre los fotograma clave como para extraer los resúmenes del video. Se construye un vector de características por cada fotograma extraído a una frecuencia de fps [4] usando el histograma de color en el espacio RGB.

Con todos estos vectores se construye una matriz sobre la cual se calcula el *Singular Value Decomposition* SVD[5]. Para obtener las características refinadas de cada fotograma con las cuales se utiliza un método de agrupamiento jerárquico para obtener los grupos que componen el resumen. En el caso de esta propuesta implica un costo computacional aunque no deja de ser interesante. Otros trabajos están dirigidos a métodos basados en la teoría de grafos, los cuales son otra forma de construir de manera jerárquica los grupos. En estos enfoques también se utiliza información estadística para obtener los diferentes niveles de resumen.

1.4 Descripción actual del dominio del problema

El SGPM desea desarrollar un software con un grupo de funcionalidades que garanticen la gestión de las medias sobre estos los archivos que se encuentran almacenados. Estos procesos pueden ser utilizados por otros proyectos que utilicen archivos multimedia, por lo que se necesita una funcionalidad que permita poder realizar resúmenes de video en cualquier instante que se requiera.

[2] Representación gráfica en forma de árbol que resume el proceso de agrupación en un análisis de clústers.
[3] Árbol cuantizador de búsqueda vectorial
[4] Fotogramas por segundo
[5] Descomposición del Valor Singular

El resumen de video se quiere hacer de forma tal que extrayendo los fotogramas claves de un material posibilite al usuario obtener un resumen del material teniendo en cuenta los datos primordiales que necesita.

La estrategia de resumen de video que se utilizará es la de lograr como salida varios resúmenes de video, con el objetivo de explotar al máximo la estructura del mismo. Sobre esta base se introduce el concepto de escalabilidad con el propósito de obtener múltiples resúmenes de video a partir de un agrupamiento jerárquico de los fotogramas claves. Este concepto resulta de gran importancia en la navegación por los fotogramas del video y dependiendo del nivel de detalles que se desee por el usuario se obtendrá mayor o menor información.

Por último, se quiere utilizar a partir de una construcción de los grupos de tomas una medida estadística, sobre la base de la duración de cada grupo y la distancia entre los fotogramas que componen el grupo. Esta medida estadística es un índice de relevancia de cada grupo el cual se utiliza para las diferentes salidas que el usuario requiere como parte del resumen del video. De manera similar se presenta una medida de relevancia con la diferencia que el autor trabaje de forma directa sobre la toma detectada. Este término de distancia para el agrupamiento jerárquico de cada fotograma extraído contiene información semántica de cada fotograma clave seleccionado. Esto permite poder construir la distancia entre los fotogramas claves como para extraer los resúmenes de video.

1.5 Caracterización de la situación problemática

Debido al gran aumento y uso de los materiales audiovisuales en el mundo, ha surgido la necesidad de hacer cada vez más eficientes las herramientas para lograr la gestión de estos. Actualmente el departamento Señales Digitales, está desarrollando el SGPM que es el que se encarga de gestionar todas las operaciones automáticas que provienen de los diferentes proyectos existentes en el departamento.

Por su parte también desarrolla una herramienta para el proyecto Catalogación y Publicación de Media (CPM), la misma debe permitir la gestión, catalogación, descripción y publicación de materiales audiovisuales digitales en la entidad que lo requiera. Esta herramienta se considera útil para la creación del catálogo digital de los materiales digitalizados que posee una institución, agilizando los procesos de búsqueda y recuperación de los mismos, necesarios en los flujos de producción de nuevos materiales audiovisuales. Además:

- Permite, la publicación de contenido audiovisual de apoyo a la docencia y los procesos fundamentales en el ámbito de la enseñanza, la descripción detallada y por ende la localización

precisa de archivos audiovisuales de los que se dispone, y la automatización de los flujos de solicitudes y préstamo de dichos documentos, logrando una mejor atención al público.

- Provee una fuente de documentación audiovisual organizada y clasificada según su tipología y que puede ser accedida en todo momento para el desarrollo de estudios o investigaciones en los que sea necesario.
- Posibilita la publicación de contenido audiovisual de interés administrativo para que esté a disposición de todos o un grupo de usuarios de la red institucional.
- Sirve de apoyo en la realización de reuniones permitiendo la transmisión de señales en vivo que pueden ser capturadas desde orígenes diversos.
- Permite a la institución que utilice la solución aplicar la clasificación de las publicaciones por secciones temáticas con diversos fines productivos, de superación, de información o entretenimiento.

El proceso de catalogación lo realizan visualizando los materiales y extrayendo la información que necesitan, permitiendo poder clasificarlos, registrarlos o etiquetarlos. Este proceso es demasiado largo, debido a que hay materiales que se demoran demasiado tiempo, por lo que el documentalista solo puede catalogar pequeñas cantidades de materiales en un mismo día.

Actualmente en el departamento existe una gran cantidad de materiales, por lo que se hace demasiado tedioso tener que visualizar todos los materiales completamente para poder obtener la información que se necesita del mismo. Realizar este proceso requiere de mucho tiempo, y se desea poder agilizar todo este proceso mostrando solo los elementos fundamentales de estos materiales y poder realizar una rápida catalogación, digitalización y publicación de las medias.

1.6 Análisis de soluciones existentes

En la actualidad existen diversas soluciones que se encargan de realizar resúmenes de videos en materiales audiovisuales y que desarrollan un profundo análisis de los mismos. Tienen como inconveniente que son propietarios y el país no puede acceder a ellos debido a los altos costos de adquisición.

1.6.1 Tarsys, de la empresa TEDIAL

Tarsys es una herramienta orientada al usuario que asegura que cualquier contenido, donde quiera que esté almacenado, pueda ser consultado por los usuarios como un único archivo. Se dedica a la gestión de

materiales audiovisuales. Permite la búsqueda de medias y la catalogación de materiales audiovisuales con información que ayude a su localización. Es capaz de elaborar resúmenes de audio y vídeo de las diferentes noticias emitidas por canales de radio y televisión para su posterior distribución a los diferentes gabinetes de prensa de los altos cargos de la administración. Cuenta con diversos módulos que pueden funcionar como productos dependientes. Estos son, *Indexer, Browser, AST e IBP Edit.*

- *Browser* es un visualizador de documentos audiovisuales y editor.
- AST se encarga de gestionar las librerías con las que se trabaja en la empresa, además de descargar parcialmente los archivos que sean necesarios.
- *IBP Edit* se encarga de editar los videos y audios para la postproducción.
- *Indexer* es un software para la indexación automática y catalogación de vídeos que incorpora un conjunto de programas que analizan los archivos multimedia y extraen de manera automática información para la catalogación. Crea un *storyboard* con los fotogramas más significativos, incorpora un conversor de audio en texto escrito, un tesauro para normalizar la indexación y facilitar la consulta, y un reconocedor de patrones (formas, colores, texturas) (Marcos, 2004).

1.6.2 Gestión de activos digitales

La solución Gestión de Activos Digitales *(DigitalAssetManagment,* por sus siglas en inglés DAM*)* de la compañía Telestream es una solución que dentro sus principales características están: lograr la extracción de metadatos y fotogramas clases para *storyboard*, permite la entrega de los metadatos extraídos al sistema DAM, genera proxy automáticos, transcodifica entre formatos, permite la reutilización de contenido para web, móvil, etc, distribuye los medios de comunicación en los sistemas de destino en los formatos requeridos por cada uno (Telestream, 2012).

1.6.3 Gestión de archivos multimedias

La solución gestión de archivos multimedias *(Media Asset Managment)* de la compañía *Cynegy* cuenta con un módulo llamado *Cynegy workspace*, *este* permite a los equipos de producción de medios colaborar en proyectos donde quiera que estén, en la oficina, en casa o en la carretera. Proporciona un acceso seguro a su base de datos desde cualquier lugar y en cualquier momento. Mediante la interfaz c*inegyworkspace* basado en el navegador, los clips se pueden buscar, navegar, seleccionar e incluso editar. Los usuarios pueden participar en un flujo de trabajo de colaboración, incluso cuando se basan en diferentes lugares. Dentro de sus principales características se encuentran: administrar de forma remota el

material. Crear, borrar, renombrar y mover carpetas y contenedores, una rápida y fácil gestión de los metadatos almacenados, presentar *storyboards* editado al convertidor *Cinegy* e integrados a través de carpetas de trabajos (Cinegy LLC, 2012).

1.6.4 Videoma

Es una solución de software para la gestión digital de video, audio, e imágenes. Realiza la ingesta de contenido en cualquier formato, catalogación, gestión y publicación del mismo. Está diseñado para empresas o entidades que necesitan gestionar su videoteca, fonoteca o fototeca de una manera organizada e incluso distribuida por internet o a través de móviles. Posee la capacidad de capturar y transcodificar cualquier tipo de formato disponible en el mercado. Ofrece la posibilidad de mantener online un archivo tanto en alta como en baja calidad, sin perder la integridad referencial de calidad por código de tiempo. Dentro de sus principales ventajas están: la posibilidad de integración con otras tecnologías del mercado, la publicacióndel contenido en un portal web, y la generación automática de *storyboard*. Cuenta también con la desventaja que es un producto privativo desarrollado sobre la plataforma de Windows (Videoma, 2012).

1.7 Metodología de desarrollo a utilizar

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un software, además es quien define, quién debe hacer qué, cuándo y cómo debe hacerlo para obtener los distintos productos parciales y finales. Existen varios tipos de metodologías entre ellas están las ágiles o ligeras y las fuertes o robustas.

Las metodologías ágiles tienen como principios el trabajo en equipo como arma fundamental; el avance del trabajo enmarcándose solamente en los elementos necesarios que este exige; la constante interacción con el cliente haciéndolo parte del equipo de trabajo; la posibilidad de cambiar todo lo que debe ser cambiado de forma que se alcance la mayor fiabilidad y calidad en el producto que se desarrolla.

Las metodologías robustas se centran en el detalle de cada proceso y de las tareas que se deben desarrollar, en las herramientas a utilizar, genera una documentación extensa que debe justificar a cada paso de avance y además adelanta la puesta en práctica de la solución. Se aplica fundamentalmente a proyectos grandes para realizar en igual período de tiempo y uso de recursos (Informática, 2009).

1.7.1 *OpenUP*

Es una versión más ágil derivada del Proceso Unificado de Desarrollo (por sus siglas en inglés, RUP), es un proceso que aplica propuestas iterativas e incrementales dentro del ciclo de vida, tratando de ser

manejable en relación con RUP. Plantea que se debe tener un software ya funcional, o lo que es lo mismo, un proyecto ejecutable en poco tiempo. Plantea que se debe utilizar sólo los procesos que sean necesarios, sin demasiados artefactos y sobre todo que el proyecto debe acoplarse a las necesidades del usuario, pudiendo ser éste modificado, mejorado y extendido.

OpenUP tiene dos ventajas importantes, este tipo de método disminuye los riesgos y además puede utilizarse tanto en proyectos pequeños como en proyectos grandes, aunque está concebida para proyectos pequeños. Si se maneja con cuidado y con profesionalismo se puede desarrollar un software de gran calidad, a pesar de que se le diseñe en poco tiempo y con poca documentación.

OpenUP contiene las características esenciales de RUP, que incluye el desarrollo iterativo de casos de uso, además de proporcionar un acercamiento a la arquitectura central del sistema. El resultado es un proceso mucho más simple que sigue fielmente los principios de RUP. OpenUP se organiza en dos dimensiones diferentes: Métodos y Procesos.

Métodos: Los roles, las tareas y los artefactos están definidos, sin tener en cuenta cómo son aplicados en el ciclo de vida del proyecto.

Procesos: Es cuando los elementos del proceso son aplicados en el sentido conductual, donde el mismo brinde los roles, las tareas y los artefactos. Pueden crearse ciclos de vida diferentes para proyectos diferentes.

Fases del ciclo de vida:

Concepción: El propósito de esta fase es lograr definir los objetivos del ciclo de vida del proyecto.

Elaboración: El propósito de esta fase es establecer una línea base arquitectónica del sistema y proveer las bases para el grueso del esfuerzo de desarrollo de la siguiente fase.

Construcción: El propósito de esta fase es completar el desarrollo del sistema basado en la arquitectura definida.

Transición: El propósito de esta fase es asegurar que el Software está listo para ser entregado a la comunidad usuaria.

Metodología de desarrollo de software seleccionada

Para darle solución a la situación problemática existente en el SGPM se utiliza la metodología de software OpenUP. Además, preserva la esencia de RUP y al igual que él, posee un modelo de desarrollo iterativo e

incremental, pero con la diferencia de que al ser un proceso mucho más ligero, permite micro incrementos en breves períodos de tiempo, lo que posibilita ir creando *raleases* del producto mientras se desarrolla y efectúa modificaciones a los requisitos sin altos costos en cuanto a tiempo, precio e implementación. Es apropiado para proyectos pequeños, lo que permite detectar errores a través de un ciclo iterativo. También, evita la elaboración de documentación, diagramas e iteraciones innecesarias requeridas por RUP y puesto que es una metodología ágil, tiene un enfoque centrado en el cliente con iteraciones muy cortas.

1.8 Tecnologías a utilizar en el desarrollo de la solución

En Cuba se hace vital el empleo de GNU\Linux para impulsar el desarrollo de la informática. Debido a esto, se hace necesario utilizar herramientas con tecnologías libres para el desarrollo exitoso de esta investigación.

1.8.1 Lenguaje de programación

Un lenguaje de programación hace referencia a un idioma artificial que está diseñado para la interacción entre un usuario y un computador, de forma tal que pueda emplearse para crear programas, y algoritmos, que solucionen problemas tales como la comunicación humana. Estos lenguajes pueden utilizarse para crear aplicaciones que cumplan con un objetivo dado. Posee procesos asociados como la depuración, la compilación, la escritura, que forman el concepto de programación. Está conformado por una serie de reglas sintácticas y semánticas que serán utilizadas por el programador y a través de las cuales creará un programa o subprograma. Por otra parte, las instrucciones que forman dicho programa son conocidas como código fuente (Lanzillotta, 2008).

Como todos los lenguajes, los lenguajes de programación poseen también sus propias clasificaciones. Por ejemplo según su nivel de abstracción se pueden clasificar en tres clases:

- Lenguaje de bajo nivel: es el código fuente de la máquina, es decir el que la máquina puede interpretar.
- Lenguaje de nivel medio: un término entre el lenguaje de la máquina y el lenguaje natural.
- Lenguaje de alto nivel: los que están compuestos por elementos del lenguaje natural, es decir el humano, especialmente el inglés.

Se pueden clasificar también atendiendo la forma en que se ejecutan, donde se pueden encontrar:

- Lenguajes compilados: programas que permiten traducir un programa del lenguaje natural al lenguaje de bajo nivel.

- Lenguajes interpretados: los que sólo hacen la traducción de los datos que se van a utilizar en ese momento y no los guarda para usar posteriormente.

C++ como lenguaje de programación

C++ es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

Posteriormente se añadieron facilidades de programación genérica, que se sumó a los otros dos paradigmas que ya estaban admitidos (programación estructurada y la programación orientada a objetos). Por esto se suele decir que el C++ es un lenguaje de programación multiparadigma.

Actualmente existe un estándar, denominado ISO C++, al que se han adherido la mayoría de los fabricantes de compiladores más modernos. Existen también algunos intérpretes, tales como ROOT.

Una particularidad del C++ es la posibilidad de redefinir los operadores, y de poder crear nuevos tipos que se comporten como tipos fundamentales.

El nombre C++ fue propuesto por Rick Mascitti en el año 1983, cuando el lenguaje fue utilizado por primera vez fuera de un laboratorio científico. Antes se había usado el nombre "C con clases". En C++, la expresión "C++" significa "incremento de C" y se refiere a que C++ es una extensión de C (Permuy, 2000).

Se escoge este lenguaje de programación debido a que se necesita un lenguaje que sea rápido, eficiente, y que brinde las funciones al equipo de desarrollo de forma tal que la construcción del software sea fluida y sin dificultades. Además, la condición de multiplataforma y multiparadigma marca una diferencia notable entre los demás lenguajes expuesto. Además porque se desea que el *plugin* se integre con el Sistema Gestor de Procesos de Medias el cual está implementado en C++ además que la biblioteca OpenCV cuenta con interfaces de C++.

1.8.2 *Framework* de desarrollo

Un *framework* tiene como objetivo ser utilizado por otros proyectos de software para ganar en organización y desarrollo. Este incluye lenguaje interpretado, soporte de programas, bibliotecas y otras herramientas para uno los componentes que se necesitan para un proyecto.

Qt como *framework* de desarrollo

Qt es un *framework* multiplataforma que se emplea en mayor medida para el desarrollo de aplicaciones con interfaz gráfica o de programas como herramientas. Puede ser utilizado tanto para realizar aplicaciones de escritorio como aplicaciones web, así como para dispositivos de tecnología móvil. Qt tiene entre sus particularidades elementos que sobresalen como son: su *framework* para aplicaciones multimedia, llamado *Phonon*; su módulo para aplicaciones 3D con *OpenGL*; el soporte para la comunicación entre procesos en tiempo de ejecución, entre otras muchas capacidades (Muñoz, 2011).

Con el objetivo de lograr el desarrollo de una aplicación que cumpla con las necesidades del cliente se utiliza como *framework* de desarrollo Qt. Una característica muy importante que maneja Qt es el paradigma señales y *slots*, también visto como eventos. Esto es un mecanismo a través del cual los objetos de una aplicación se comunican. Este mecanismo es una de las características distintivas de Qt. Además de brindar facilidades de uso, este *framework* posee una documentación extensa que posibilita el entendimiento, aprendizaje y desarrollo de su utilización para los usuarios. En la UCI existen grupos de desarrollo que utilizan Qt para todos los procesos que realizan, esto garantiza que exista una documentación extensa sobre el mismo.

Como último elemento se puede plantear que la portabilidad que posee Qt permite que pueda funcionar en diversas plataformas ya sean sistemas tipo *Unix* con el servidor gráfico *X Windows System* (Linux, BSDs, Unix), para *Apple Mac OS X*, para sistemas de *Microsoft Windows*, para Linux Embebido e incluso para dispositivos que utilizan *Windows CE*. Dependiendo del entorno donde se haga la compilación, el módulo *qmake* hará su función y generará los archivos necesarios para que el programa pueda ser ejecutado en el entorno donde se está trabajando sin necesidad de adaptar el código (Ramiro, 2011).

1.8.3 Biblioteca a utilizar

Una biblioteca es un conjunto de subprogramas utilizados para desarrollar software. Las bibliotecas contienen código y datos, que proporcionan servicios a programas independientes, esto permite que el código y los datos se compartan y puedan modificarse de forma modular. Algunos programas ejecutables pueden ser a la vez programas independientes, pero la mayoría de éstas no son ejecutables. La mayoría de los sistemas operativos modernos proporcionan bibliotecas que implementan la mayoría de los servicios del sistema.

Open Source Computer Vision (OpenCV)

OpenCV, es una biblioteca abierta desarrollada por Intel. La misma facilita varias funciones para el procesado de imágenes permitiéndoles a los programadores crear aplicaciones poderosas en el dominio

de la visión digital, ofrece cuantiosos tipos de datos de alto-nivel como juegos, árboles, gráficos, matrices, etc. Además permite operaciones básicas, procesado de imágenes, análisis estructural, análisis de movimiento, reconocimiento del modelo, reconstrucción 3D y calibración de la cámara, interfaz gráfica y adquisición, etc.

OpenCV implementa una gran diversidad de herramientas para la interpretación de la imagen. Es compatible con *Intel ImageProcessing Library* (IPL) que implementa algunas operaciones en imágenes digitales. A pesar de primitivas como binarización, filtrado, estadísticas de la imagen, pirámides, *OpenCV* es primordialmente una biblioteca que implementa algoritmos para las técnicas de la calibración (Calibración de la Cámara), detección de rasgos, para rastrear (Flujo Óptico), análisis de la forma (Geometría, Contorno que Procesa), análisis del movimiento (Plantillas del Movimiento, Estimadores), reconstrucción 3D (Transformación de vistas), segmentación de objetos y reconocimiento (Histograma, etc.).

El rasgo fundamental de la biblioteca junto con funcionalidad y la calidad es su desempeño. Los algoritmos están asentados en estructuras de datos muy flexibles, conectados con estructuras IPL; más de la mitad de las funciones ha sido perfeccionada beneficiándose de la Arquitectura de Intel (Adrian Kaehler, 2008).

Esta biblioteca se divide en varios módulos:

- CORE: donde se encuentran las estructuras y algoritmos básicos que usan las demás funciones tales como: suma, media, operaciones-binarias.
- Imgproc: Posee todas las funciones de procesamiento sobre las imágenes.
- HighGUI: Esta módulo contiene funciones valiosas sobre cómo leer / guardar la imagen / archivos de vídeo.
- Calib3D: Se encarga de encontrar a partir de las imágenes 2D información sobre el mundo 3D.
- Video: En este módulo se puede encontrar el empleo de algoritmos stream de video como: extracción de movimiento, función de seguimiento y extracción de primer plano.
- Objdetect: Módulo que se encarga de detectar los objetos.

- ML[6]: Módulo que posee las clases de aprendizaje de máquinas potentes para la clasificación estadística, la regresión y la agrupación de los datos.
- GPU: Módulo que se encarga de exprimir cada una de las pequeñas potencias de cálculo del sistema, utilizando el poder de la tarjeta de vídeo para ejecutar el algoritmo de OpenCV.

Las que se utilizarán en esta investigación serán: CORE, HighGUI, debido a que en estos dos módulos es donde se encuentran las principales funcionalidades para realizar el tratamiento con las imágenes del video, así como poder cambiarle el color, poder calcular el histograma de color, entre otras funcionalidades.

1.8.4 Entorno integrado de desarrollo (por sus siglas en inglés IDE)

Luego de haber escogido un *framework* de desarrollo y un lenguaje de programación y debido al avance de la investigación, se brinda la posibilidad de escoger un IDE de programación, para la selección del IDE se establecen parámetros a cumplir: compatibilidad con GNU\Linux, soporte para el lenguaje C++, buen completamiento de código, integración con bibliotecas de Qt, consumo de memoria reducido y por último abundante documentación.

QtCreator es un IDE multiplataforma creado por *Trollthech* para el desarrollo de aplicaciones con bibliotecas Qt y distribuido por Nokia como parte del Qt SDK (equipo de desarrollo de software de Qt), que es un grupo de herramientas para desarrollar con el uso del *framework* Qt. Posee un avanzado editor de código C++, lenguaje que utiliza de forma nativa. Al utilizar el lenguaje de programación C++, se hace uso de la programación orientada a objetos. QtCreator posee herramientas como: *Qt Assistant*, que permite interactuar con la documentación de Qt, donde se muestran todos los elementos que se deben dominar para el trabajo con el IDE; *Qt Designer*, permite crear las interfaces graficas de usuario; *Qt Linguist*, permite crear aplicaciones con soporte para varios idiomas. *Qt Creator* puede generar un paquete de instalación cuando un dispositivo móvil se conecta a un PC, al conectarlo se crea un código y lo ejecuta (Ramiro, 2011).

Este brinda las siguientes prestaciones:

- Diseño de interfaz de usuario integrado.
- Proyecto y construcción de herramientas de gestión.
- Soporte para el control de versiones.

[6] Del inglés *machine learning-aprendizaje automático*

- Creación de bibliotecas.
- Implementación de señales y slots.

Se decide utilizar QT como IDE de desarrollo, debido a que se desea integrar este componente con el SGPM y el mismo está trabajando con este IDE de desarrollo. Además, el mismo brinda una amplia documentación que permite conocer más sobre su funcionamiento (Muñoz, 2011).

1.9 Herramienta CASE para el modelado

Las herramienta CASE son programas y ayudas que dan asistencia a los analistas, ingenieros de software, durante todo el proceso de ciclo de vida del desarrollo de un software (Alfaro; Félix Murillo, 1999).

Con estos tipos de herramientas se puede diseñar, implementar a partir de un diseño realizado, compilar automáticamente, detectar errores y brindarle la seguridad al equipo de trabajo sobre el avance de la solución.

Visual Paradigm para UML

Dependiendo de los requisitos que debe cumplir la herramienta CASE a utilizar en el desarrollo se ha decidido emplear Visual Paradigm para UML. Es una herramienta profesional que soporta el ciclo de vida del desarrollo del software y posibilita un ahorro considerable de tiempo y una calidad óptima en el proceso. Se considera muy completa y fácil de usar, con soporte multiplataforma y excelentes facilidades de interoperabilidad con otras aplicaciones. Posibilita la captura de requisitos, análisis, diseño e implementación para una aplicación en desarrollo.

Entre las ventajas que ofrece están (Sierra, 2006):

- Soporte para UML versión 2.1: con la selección de la metodología OpenUP esta característica es muy provechosa ya que se necesita modelar los diagramas con el uso de UML.
- Generación de código: Modelo a código, diagrama a código, para diferentes lenguajes, entre ellos C++, Java y exportación como HTML.
- Generación de bases de datos: permite la generación automática de bases de datos a partir de un Modelo entidad-relación.
- Interoperabilidad entre diagramas: permite a partir de un diagrama obtener otro que guarde relación con el mismo.
- Tiene apoyo adicional en cuanto a generación de artefactos automáticamente.

- Disponibilidad en múltiples plataformas: *Microsoft Windows* (98, 2000, XP, o Vista), *Linux*, *Mac OS X, Solaris o Java.*
- Generación de documentación: brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.

Lo anteriormente expuesto demuestra que *Visual* Paradigm es una buena opción alternativa de herramienta CASE para un desarrollo exitoso del componente que se implementa ya que agilizará el proceso de desarrollo y generará los estereotipos necesarios con la estructura y relaciones deseada.

1.10 Conclusiones parciales

Luego de haber analizado los proyectos de CPM y SGPM, se concluye que para procesar los materiales audiovisuales se requiere de una herramienta que permita realizar resúmenes de videos.

El estudio del estado del arte sobre otros sistemas similares evidencia la inexistencia de aplicaciones libres, o sistemas que no son aplicables al entorno del departamento Señales Digitales. El planteamiento de la fundamentación teórica aportó los elementos esenciales que guiarán el desarrollo de la investigación, dejando así plasmados los conceptos asociados al dominio del problema en cuestión.

En la presente investigación se utilizará OpenUP como metodología de desarrollo. Teniendo en cuenta las políticas del departamento Señales digitales y las particularidades de la solución propuesta se decide utilizar el lenguaje de programación C++, *framework* Qt, el IDE Qt *Creator*, y OpenCV como biblioteca ,además de Visual Paradigm como herramienta CASE para el modelado.

Capítulo 2: Características del componente

En el capítulo se exponen las principales características del sistema a implementar mediante la especificación de los requisitos funcionales y no funcionales con los que debe cumplir la solución, además se describe detalladamente el algoritmo a seguir para dar solución al problema que dio origen a la presente investigación.

2.1 Descripción del algoritmo

En la presente investigación se desarrolla un componente que debe generar *storyboards* a partir de un video, permitiendo al departamento Señales Digitales poder agilizar todas las operaciones relacionadas con las medias. El componente desarrollado se integrará con el SGPM como un *plugin* a través de un sistema de *plugins* que maneje sus funciones como procesos de un mismo sistema.

Se propone realizar un *plugin* que solucione la deficiencia de tener que visualizar completamente los materiales para poder extraer toda la información que necesiten de los mismos para así poder realizar las operaciones de descripción, y catalogación en los proyectos que así lo requieran. Para una mejor integración del componente desarrollado se aplica la programación mediante hilos de procesos ya que esta funcionalidad puede ser requerida por varios sistemas y él debe de saber cuál petición debe atender primero, realizando así una ejecución proceso a proceso.

Se propone para obtener el resultado final de esta investigación seguir el orden mostrado en la siguiente figura (Figura 1).

Figura 1: Descripción del algoritmo

Como se demuestra en la figura (Figura 1), a partir de un video original se extraen los fotogramas, los cuales son la entrada al método de detección cambio de tomas. Para el proceso de decodificación se utilizaron los métodos que posee la librería OpenCV para lograr el acceso a los fotogramas del video. Después de la decodificación del video se ejecutan los métodos de detección de cambio de tomas los

cuales se dividen en dos partes fundamentales. En primer lugar la función de distancia que determina la similitud entre los fotogramas consecutivos y luego el proceso de existencia o no de cambio de tomas. Este método después de procesar los fotogramas del video, se obtiene como salida una lista de índices donde se encuentran los cambios de tomas del video original. A partir de estos índices se seleccionan los fotogramas claves. El paso siguiente es hacer un agrupamiento para unificar aquella toma que son cercanas en el tiempo y contienen coherencia en la información semántica que poseen estos fotogramas claves. Luego de haber extraído los fotogramas claves se pasa a la construcción del resumen del video.

Decodificar video

Para la decodificación del video se utilizan los métodos de lectura de video que posee OpenCV. Para ello es necesario crear una estructura de OpenCV de tipo Capture y con la función VideoCapture& VideoCapture**::**operator>> se accede de manera consecutiva a los fotogramas que componen el video. Para el método detectar cambio de toma se hace necesario la información de dos fotogramas consecutivos, es necesario en la decodificación extraer los dos fotogramas. En este sentido, antes de iniciar el recorrido completo del video es necesario extraer el primer fotograma para luego en la primera corrida del ciclo extraer el siguiente y a partir de este momento en cada recorrido usar el fotograma actual y el consecutivo.

Detección cambio de toma

El proceso de detección de cambio de toma se modeló a partir de un vector de distancia entre fotogramas consecutivos de un video en el cual aparecen fronteras de las tomas como picos pronunciados en dicha función. La figura 2 muestra la representación de un vector de distancia con la información visual asociada se refleja en la siguiente figura, los cambios visuales representados por picos en la función de distancia.

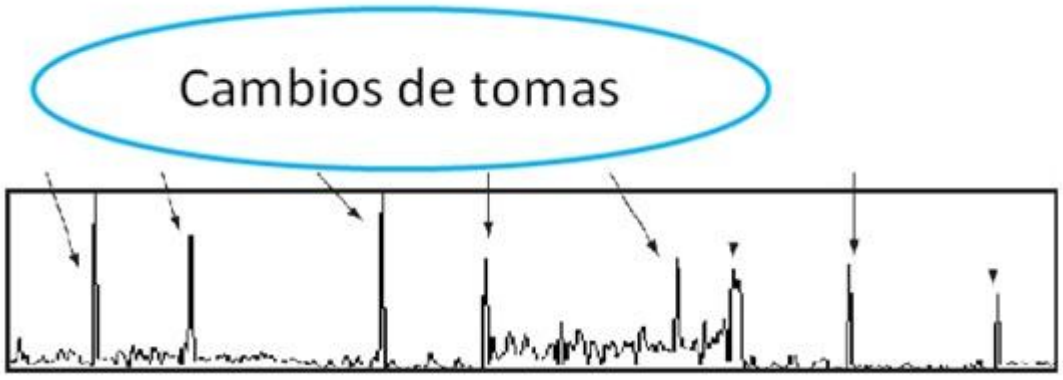

Figura 2: Detección de cambio de toma

Luego, se hace un análisis del método de distancia más utilizado siendo este el método basado en histogramas de colores.

Método calcular distancia

Para el proceso de detectar cambio de toma, primeramente debe tenerse en cuenta la similitud que existe entre los fotogramas de una misma toma. El método propuesto para calcular esta distancia entre dos fotogramas consecutivos del video es basado en los histogramas de colores, debido a que desde el punto de vista teórico el histograma de color de una imagen es una característica global de la imagen que identifica la distribución de colores en la misma. Esta dada por la siguiente ecuación:

$$h_{A,B,C} = N\,Prob(A = a, B = b, C = c)$$

Donde A,B,C son los valores que representan los tres canales de color (RGB o HSV) y N es el número de píxeles de la imagen. Por ser una medida global, resulta muy difícil que esta cambie de manera significativa en las imágenes que componen una misma toma. En la propuesta presentada se utilizó el espacio de colores HSV[7], se propone este espacio de color ya que a diferencia del RGB, este es un espacio de color perceptualmente uniforme

Pasos para calcular la distancia entre dos fotogramas consecutivos basado en histograma de color:

1. Se convierte la primera imagen al espacio de color HSV.
2. Se calcula el histograma de color para esta imagen, utilizando el método que posee opencv, cv::calchist.
3. Se convierte la segunda imagen al espacio de color HSV y se determina el histograma de color.
4. Se determina la distancia que está dada por la siguiente ecuación utilizando el método que posee opencv:

$$d = 1 - cv :: compareHist(his1, his2, \text{CV\_COMP\_CORREL})$$

El método cv::compareHist compara dos histogramas, utilizando el método específico que le pases por parámetro. En este caso se utilizó el método CV_COMP_CORREL o método de correlación, dado por la siguiente ecuación:

[7] del inglés *Hue, Saturation, Value* – Matiz, Saturación, Valor

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H_1})(H_2(I) - \bar{H_2})}{\sqrt{\sum_I (H_1(I) - \bar{H_1})^2 \sum_I (H_2(I) - \bar{H_2})^2}}$$

Dónde:

$$\bar{H_k} = \frac{1}{N} \sum_J H_k(J)$$

Siendo N el número total de posiciones del histograma.

El proceso siguiente una vez determinada la distancia entre los fotogramas consecutivos es lograr ajustar la distancia para reducir el ruido que pueda aparecer en algún segmento de la señal de distancia cuyos valores pueden estar afectados por algún movimiento en el contenido del video y que realmente no se está en presencia de un cambio de toma.

Ajustar la función de distancia

Antes de aplicar el proceso de detección de cambios de tomas se propone utilizar un método local de ajuste de la distancia de los fotogramas del video. Esta función de ajuste se propone el uso de una ventana deslizante de tamaño 2w+1. Para realizar el ajuste por cada valor de la lista de distancias se toman los valores de distancia de los w vecinos hacia la izquierda y hacia la derecha del valor que se analiza. Se determina la media de todos estos valores para finalmente calcular el valor absoluto de la distancia que se analiza restado a la media de los valores de la ventana. La siguiente expresión describe de manera formal la estrategia antes mencionada.

$$\boldsymbol{d_1(i) = \left| d(i) - \frac{1}{2w+1} \sum_{k=i-w}^{k=i+w} d(k) \right|}$$

Siendo d(k) la distancia del fotograma que este en esa posición en ese momento y W se utilizó con valor igual a 2.

Detectar cambio de tomas

El hecho de que un resumen de video no requiere de gran exactitud en los métodos de detección de cambios de tomas, en varios trabajos (Benini, 2006) (Chen, 2010) existe un enfoque de agrupamiento que

hace que la estructura inicial de separación en tomas se pierda pasando a una estructura de escenas. Por tal motivo se propone emplear un método de umbral global automático.

Este método de umbral se aplica sobre la función de distancia ajustada. El cálculo automático del umbral se basa en determinar la media (m) y la varianza (s) de todos los valores de la función de distancia ajustada y con ello construir un umbral $T = m + s$ que nos permita determinar aquellos valores que sean superiores a este umbral como posibles cambios de tomas.

En el enfoque se habla de posibles cambios de tomas pues luego de aplicar el umbral se hace un pos-procesamiento que se basa en el principio de que no existen dos cambios de tomas consecutivos cuya diferencia de tiempo sea inferior a un valor predeterminado. Por lo que se recorren todos los posibles cambios de tomas se define una ventana de tiempo fija y se verifica que no existan dos o más tomas consecutivas dentro de esta ventana. De cumplirse esta condición se selecciona como cambio de toma al mayor de todos los posibles cambios de tomas dentro de la ventana y de no cumplirse la condición simplemente se etiqueta como cambio de toma.

Selección de fotogramas claves

Los métodos de detección de fotogramas permiten a partir de una imagen fija representar el contenido de una toma. Para ello se utilizan con frecuencias métodos simples que emplean el centro de la toma o los extremos de la toma como fotograma clave. Esto a pesar de ser simple no se considera para nada inefectivo pues dentro de una misma toma los fotogramas que la componen no presentan diferencias significativas. El método de selección de fotogramas claves propuesto utiliza el centro de la toma.

Agrupamiento jerárquico

Para la selección de los fotogramas representativos se propone emplear un método de agrupamiento jerárquico basado en el criterio de la distancia mínima entre los fotogramas claves.

Los pasos que se siguen para el agrupamiento son fundamentalmente los siguientes:

1. Calcular la distancia entre los fotogramas claves consecutivos de un video.
2. Agrupar el par de fotogramas claves de menor distancia en un clúster[8].

[8] Clúster: grupo de lo mismo o similares elementos recogidos.

3. Crear un nuevo clúster y asignarle como sus hijos los clústeres seleccionados en el paso 2 recalculando la distancia con el nuevo clúster a partir de los valores medios de las distancias anteriores.

4. Poner el nuevo clúster dentro del listado general y eliminar sus hijos de este listado.

5. Repetir desde el paso 2 hasta que se tenga un solo clúster padre de todos.

Luego de concluir estos pasos, se contará con un clúster que dependiendo el nivel de profundidad que se solicite, será capaz de dar los fotogramas representativos del video.

Exportar los fotogramas como imagen

Al concluir el proceso de extracción de fotogramas representativos, estos se deben guardar en una dirección local, estos fotogramas están identificados por su posición dentro del video por lo que se debe calcular el tiempo exacto en el que se encuentran en milisegundos, para poder exportarlo en el formato "hh_mm_ss".

$$MSC = \frac{1000 * PF}{Fps}$$

En la fórmula anterior se calcula el milisegundo (*MSC*) donde se encuentra un fotograma, teniendo la posición del fotograma (*PF*) y la cantidad de fotogramas por segundo con que se debe reproducir el video.

Después de tener los milisegundos exactos en que se encuentran los fotogramas representativos, se debe salvar la imagen haciendo una conversión de tiempos, es decir de milisegundos a horas, minutos y segundos, para darle origen al nombre de la imagen.

Algunos elementos del proceso de integración:

La implementación del presente componente tiene como objetivo su futura integración con el producto SGPM, de aquí la necesidad de definir un conjunto de parámetros de entradas y salidas para su correcto funcionamiento e integración. Como parámetros de entrada se definieron los siguientes:

Dirección de entrada: Es la dirección física de la media a la que se le desea extraer los fotogramas claves.

Dirección de salida: Lugar de la PC en que se desean salvar los fotogramas claves.

Sensibilidad: Representa el porciento de resumen que se desea obtener luego de aplicar el proceso de segmentación al video.

Como parámetro de salida se definió el siguiente:

Porciento de ejecución en que se encuentra el proceso. El componente debe informar a la plataforma en cada momento en qué porciento del proceso de resumen se encuentra.

2.2 Modelo del dominio

Un modelo del dominio se utiliza con frecuencia como fuente de inspiración para el diseño de los objetos del software. El modelo del dominio muestra, a los modeladores, clases conceptuales significativas en el dominio del problema, es el artefacto más importante que se crea durante el análisis orientado a objetos.

Un modelo del dominio se representa con un conjunto de diagramas de clases en los que no se define ninguna operación. Pueden mostrar:

- Objetos del dominio o clases conceptuales.
- Asociaciones entre las clases conceptuales.
- Atributos de las clases conceptuales.

Diagrama de modelo del dominio

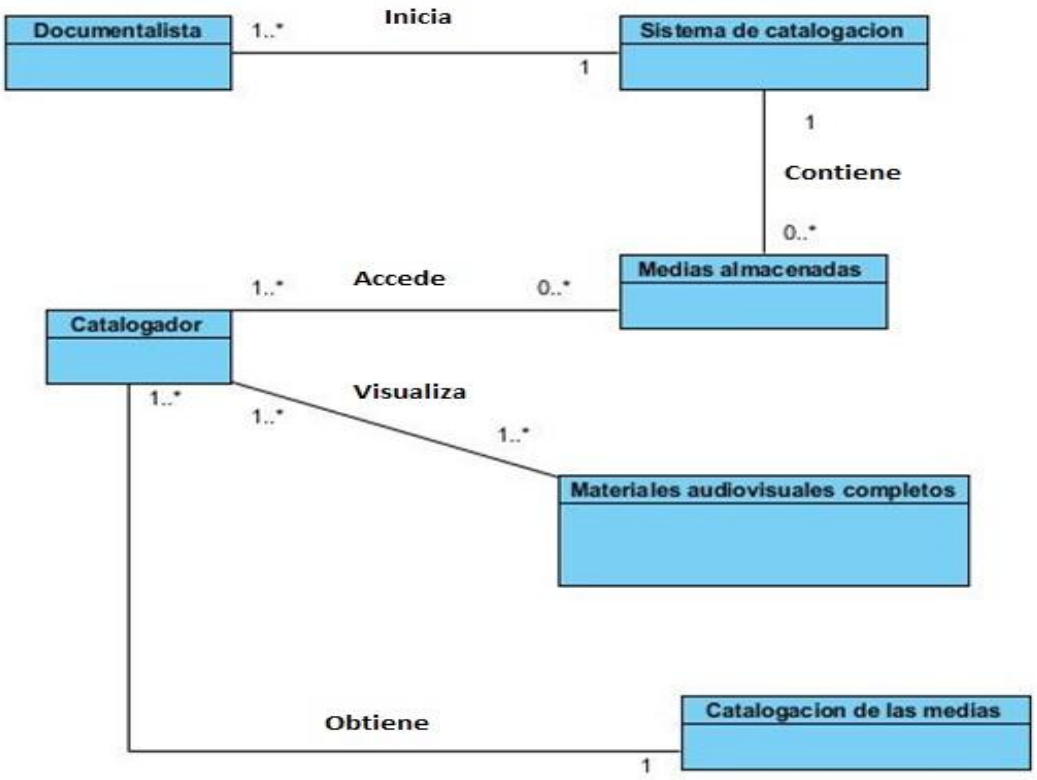

Figura 3: Modelo del dominio

Descripción del Modelo del Dominio

El modelo del dominio describe el proceso de identificación de fotogramas claves en dos momentos: cuando este es realizado a través del Sistema de Catalogación (SC) y cuando es llevado a cabo a través de la visualización de un material audiovisual.

- En el primer caso se tiene un documentalista que accede a trasvés del SC a un video (media) almacenado con anterioridad para registrar todas las posibles informaciones que necesiten para el proceso de catalogar. Considerando esta acción como un tipo de identificación del video.
- En el segundo caso un catalogador puede acceder a las medias directamente o controlar el proceso de visualización de un material. En ambos casos mientras se lleva a cabo el proceso de visualización del material se documenta el registro de fotogramas claves con los que se cuenta en el mismo.

1.3 Descripción de las clases del Modelo del Dominio

Documentalista: A través del sistema de catalogación inicia una media para obtener los datos que necesita.

Sistema de catalogación: Brinda una plataforma para el trabajo del documentalista y contiene las medias almacenadas.

Medias: Son capturas de video almacenadas para ser consultadas por documentalistas o el catalogador.

Catalogador: Se encarga de dar seguimiento al contenido de una media para identificar los momentos de cambios de toma. Posteriormente se encarga de extraer los fotogramas claves que necesita para poder realizar la catalogación.

Materiales visuales completos: Es el medio a través del cual el catalogador es capaz de obtener la información necesaria.

Catalogación de las medias: Es el documento que contiene toda la información extraída de los videos.

2.3 Identificación de los requisitos

Para darle solución al problema planteado, primeramente se debe conocer los requisitos funcionales con los que debe cumplir el componente que se propone, los cuales fueron detectados a lo largo de la investigación.

2.3.1 Requisitos funcionales del componente

Los requisitos funcionales muestran cuál será el comportamiento interno del sistema que se desarrolla. Son las condiciones que debe tener el sistema. A continuación se muestran los definidos en esta investigación:

RF1:Detectar cambio de tomas.

Descripción: El sistema propuesto debe ser capaz de dada la dirección de un video detectar los cambios de tomas existentes en el mismo.

Entradas:

- Video decodificado.

Salidas:

- Intervalos de tiempo donde se realizaron los cambios de tomas.

RF1.1:Decodificar video.

Descripción: El sistema debe permitir decodificar un archivo de video del cual se conoce su dirección de origen.

Entradas:

- Dirección origen del archivo de video sobre el cual se va decodificar para obtener la información deseada por el sistema.

Salidas:

- Cadena contenedora de la información del video según los parámetros indicados por el sistema.

RF1.2: Calcular distancia por histogramas de color

Descripción: Permite obtener la distancia existente entre dos imágenes, convirtiendo las imágenes al espacio de colores HSV.

Entradas:

- Imágenes de las que se desea conocer la distancia.

Salidas:

- Distancia existente entre estas dos imágenes.

RF2: Escoger fotogramas representativos dentro de los cambios de tomas.

Descripción: El sistema propuesto debe ser capaz de reconocer los fotogramas representativos dentro de las tomas de un video, así como agruparlos por el factor de similitud que posean entre ellos, para reducir el número de los mismos.

Entradas:

- Cambios de tomas detectados.

Salidas:

- Fotogramas claves.

RF3:Exportar como imagen JPEG los fotogramas representativos.

Descripción: El componente propuesto debe ser capaz de exportar los fotogramas representativos de un video a una dirección local dada. Cumpliendo con los siguiente: las imágenes se guardaran como JPEG y el nombre de las imágenes coincidirá con la aparición en tiempo que poseen dentro del video, cumpliendo con el siguiente formato”hh_mm_ss”[9]

Entradas:

[9]Horas_Minutos_Segundos

- Dirección origen del archivo de video sobre el cual se va a interactuar para obtener toda la información deseada por el sistema.
- Cadena de parámetros que indiquen qué atributos del video hay que extraer.

Salidas:

- Cadena contenedora de la información del video según los parámetros indicados por el sistema.

2.3.2 Requisitos no funcionales del componente

Los requisitos no funcionales son las características que debe de tener el sistema que se desarrolla. Existen muchas formas de clasificar estos requisitos por categorías. A continuación se muestran los definidos por esta investigación:

Software:

- Se requiere la biblioteca OpenCV versión 2.2 o superior.

Restricciones de diseño y de implementación:

- Para el desarrollo se emplearán sólo herramientas libres, como el IDE Qt Creator versión 2.2.1, las bibliotecas especificadas de Qt versión 4.8 y la biblioteca OpenCV para el procesado de las imágenes.

2.4 Descripción del sistema propuesto

Diagrama de caso de uso del sistema

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar. Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente (Tello, 2012).

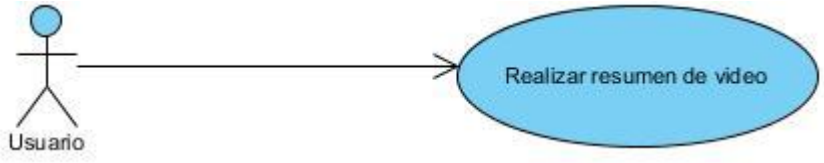

Figura 4 : Diagrama de CUS

Actor:

Usuario: Es el encargado de poder realizar el resumen de video de solicitar la opción de realizar resumen de video al SGPM.

Especificación de Casos de uso

Tabla 1 Caso de uso "Realizar resumen"

Objetivo	**Realizar resumen de video**	
Actores	Usuario	
Resumen	El *plugin* propuesto debe generar resumen de video en forma de *Storyboard*.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Debe estar disponible la aplicación del SGPM Debe estar integrado el *plugin* ”resumen de video”	
Postcondiciones	Se obtiene el resumen de video	
Flujo de eventos		
Flujo básico: Realizar resumen de video.		
	Actor	**Sistema**
	1. Solicita la opción realizar resumen de video o sumarizacion de video.	
		2. Se muestra una ventana con los parámetros de dirección del video, y donde desea guardar las imágenes una vez realizado el resumen.
	3. Se introducen los datos y se selecciona la opción “Comenzar”.	4. En caso que los datos estén correctos se realiza el resumen de video, en caso contrario ir al paso 4.1

		5. Fin del caso de uso.
Flujo Alternos		
Flujo básico: Realizar resumen de video.		
	Actor	**Sistema**
		4.1 El sistema verifica que todos los datos estén correctos, en caso que no lo estén envía un mensaje de error notificándoselo al usuario.

Diagrama de colaboración

En los diagramas de colaboración se muestran las interacciones entre objetos creando enlaces entre ellos y añadiendo mensajes a esos enlaces. El nombre de un mensaje debería denotar el propósito del objeto invocante en la interacción con el objeto invocado. (James, 2005)

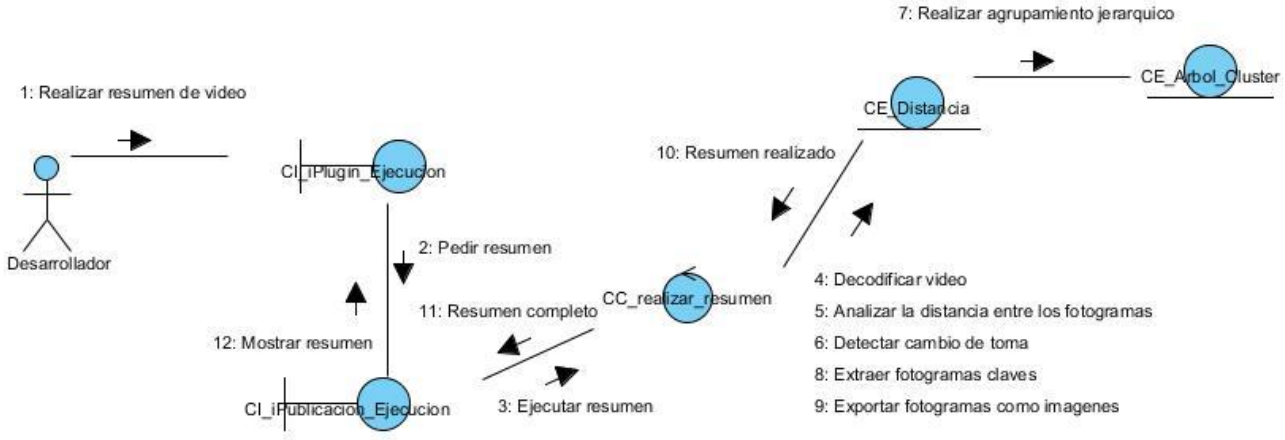

Figura 5: Diagrama de colaboración

Diagrama de secuencia

Un Diagrama de Secuencia muestra los objetos de un escenario mediante líneas verticales y los mensajes entre objetos como flechas conectando objetos Los mensajes son dibujados cronológicamente desde arriba hacia abajo Los rectángulos en las líneas verticales representan
los periodos de actividad de los objetos (Software Engeneering Lab, 2013).

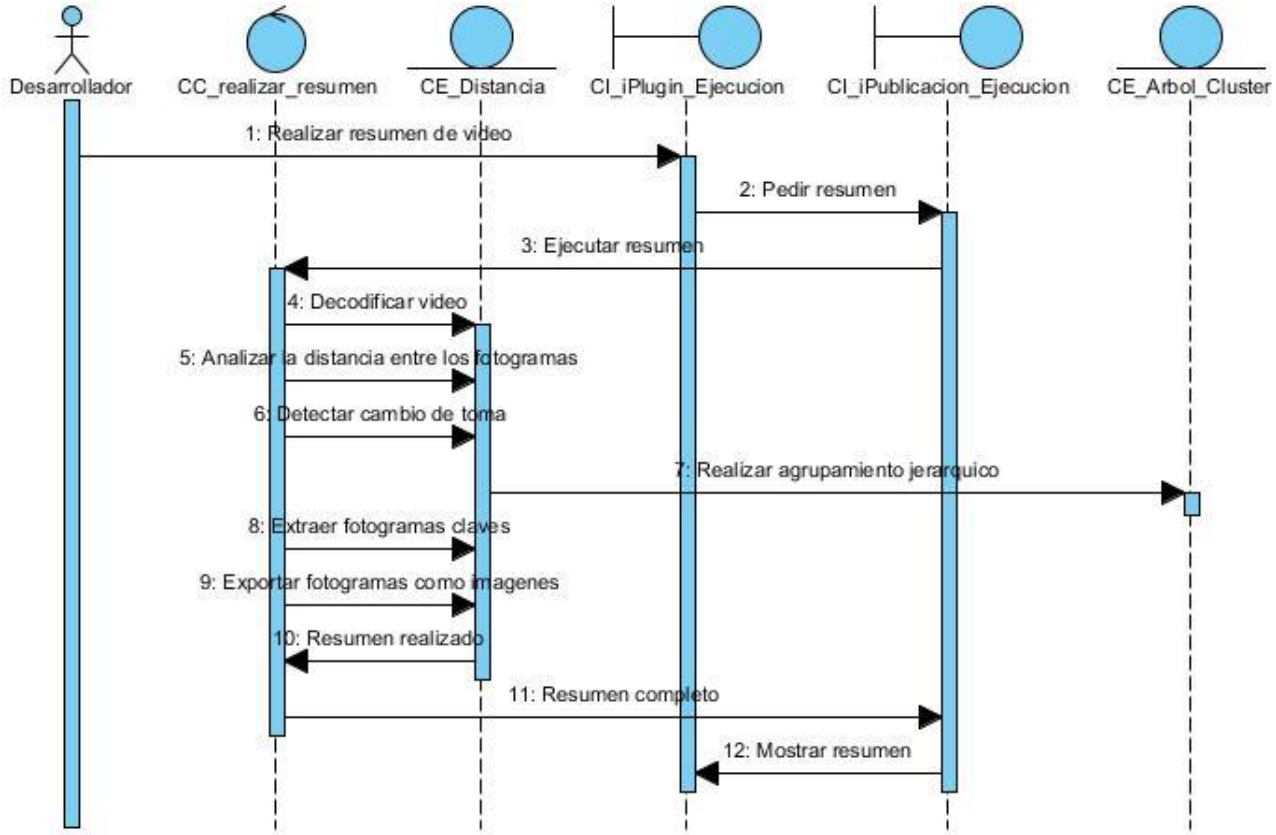

Figura 6: Diagrama de secuencia

2.5 Arquitectura propuesta

La arquitectura de software es de especial importancia ya que es la manera en que se estructura un sistema y tiene un impacto directo sobre la capacidad de este para satisfacer lo que se conoce como los atributos de calidad del sistema (Bass, 2008).

Tuberías y filtros:

La realizar de resumen de video se basó en el estilo arquitectónico llamado *Pipes & Filters* (Tuberías y filtros), donde los datos de entrada se han de transformar en datos de salida. Es un componente que lee un flujo de datos en la entrada y produce un flujo de datos diferentes en su salida. Esto es logrado aplicando una trasformación local al flujo de entrada mientras este se lee, de tal forma que el flujo de salida empieza antes que se consume todo el flujo de entrada. Este patrón divide la tarea de un sistema en varios pasos de procesamiento secuencial. Estos pasos están conectados por el flujo de datos a través del sistema, cada paso del procesamiento esta encapsulado en un componente de filtro. Los datos pasan

a través de las tuberías que son los conectores que sirven como conducto para transmitir las salidas de un filtro a las entradas de otro.

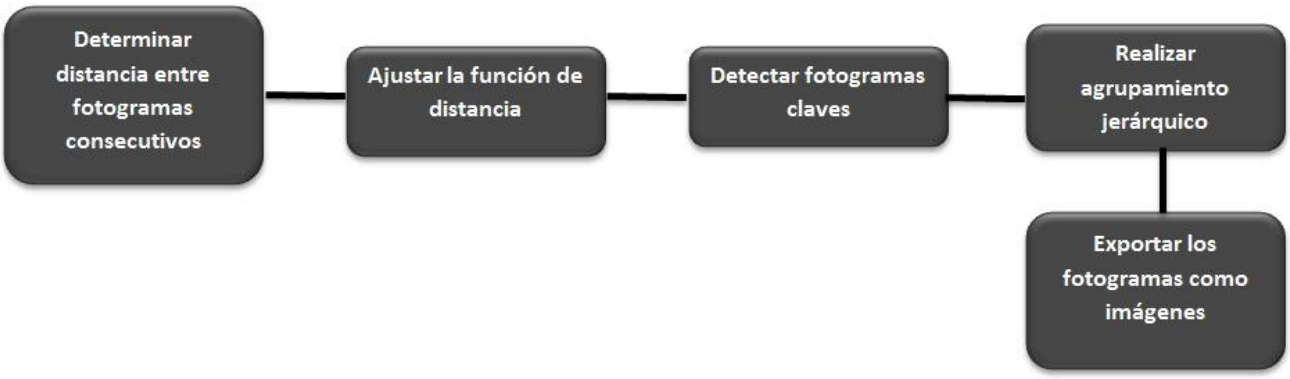

Figura 7: Estilo arquitectónico

Este patrón de arquitectura es particularmente efectivo a la hora de descomponer el problema en varios pasos independientes, reutilizar filtros, facilitar el mantenimiento, independencia y ejecución concurrente de filtros. Este patrón tiene algunas restricciones, como que los filtros deben ser entidades independientes: en particular no deben compartir estados con otros filtros. Los filtros no conocen la identidad del filtro de donde proviene el flujo que reciben como entrada y tampoco la identidad del filtro a donde llega el flujo de salida.

2.6 Modelo del diseño

Diagrama de Clase del diseño

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

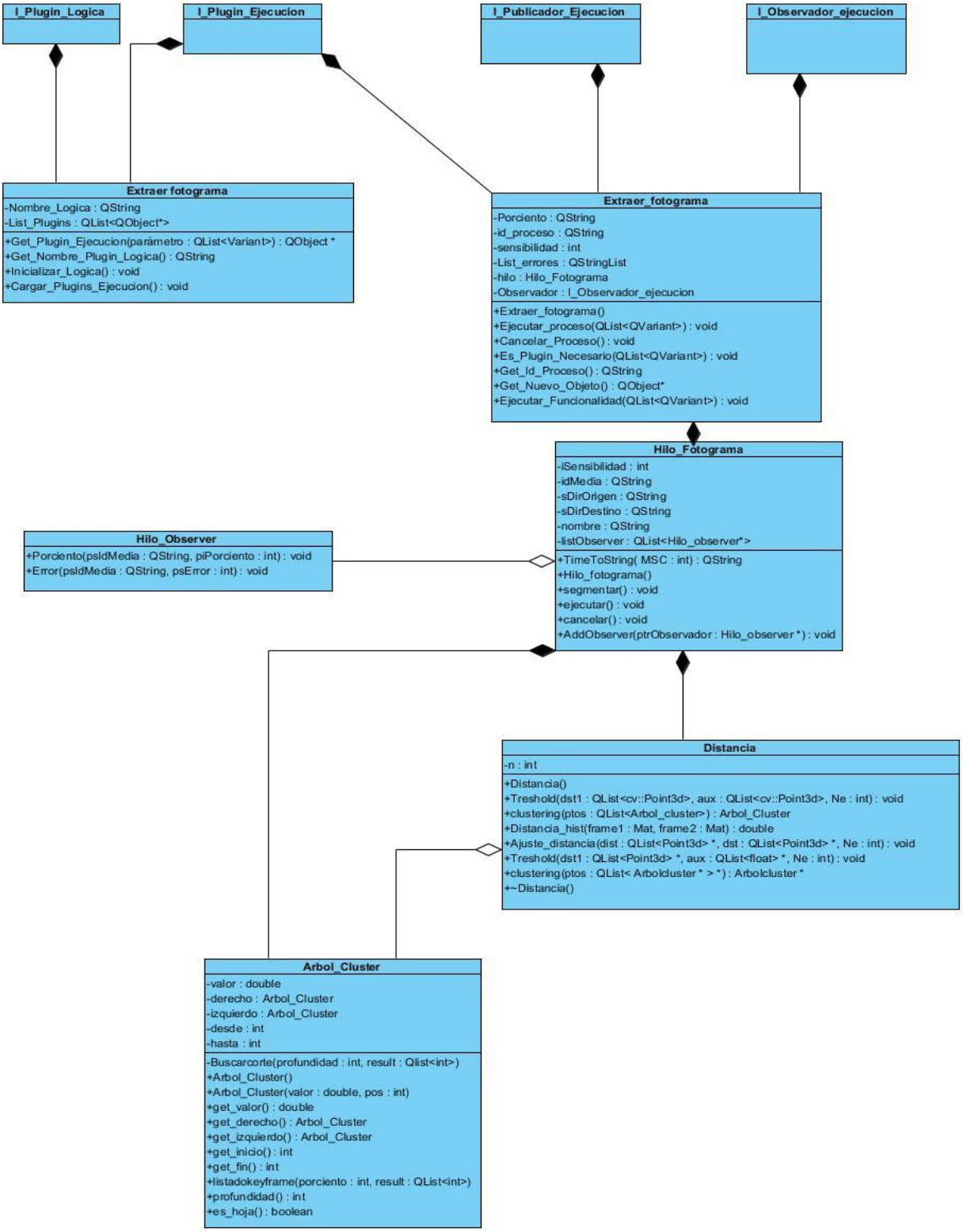
I_Plugin_Logica
I_Plugin_Ejecucion
I_Publicador_Ejecucion
I_Observador_ejecucion
Extraer fotograma
-Nombre_Logica : QString
-List_Plugins : QList<QObject*>
+Get_Plugin_Ejecucion(parámetro : QList<Variant>) : QObject *
+Get_Nombre_Plugin_Logica() : QString
+Inicializar_Logica() : void
+Cargar_Plugins_Ejecucion() : void
Extraer_fotograma
-Porciento : QString
-id_proceso : QString
-sensibilidad : int
-List_errores : QStringList
-hilo : Hilo_Fotograma
-Observador : I_Observador_ejecucion
+Extraer_fotograma()
+Ejecutar_proceso(QList<QVariant>) : void
+Cancelar_Proceso() : void
+Es_Plugin_Necesario(QList<QVariant>) : void
+Get_Id_Proceso() : QString
+Get_Nuevo_Objeto() : QObject*
+Ejecutar_Funcionalidad(QList<QVariant>) : void
Hilo_Fotograma
-iSensibilidad : int
-idMedia : QString
-sDirOrigen : QString
-sDirDestino : QString
-nombre : QString
-listObserver : QList<Hilo_observer*>
+TimeToString(MSC : int) : QString
+Hilo_fotograma()
+segmentar() : void
+ejecutar() : void
+cancelar() : void
+AddObserver(ptrObservador : Hilo_observer *) : void
Hilo_Observer
+Porciento(psldMedia : QString, piPorciento : int) : void
+Error(psldMedia : QString, psError : int) : void
Distancia
-n : int
+Distancia()
+Treshold(dst1 : QList<cv::Point3d>, aux : QList<cv::Point3d>, Ne : int) : void
+clustering(ptos : QList<Arbol_cluster>) : Arbol_Cluster
+Distancia_hist(frame1 : Mat, frame2 : Mat) : double
+Ajuste_distancia(dist : QList<Point3d> *, dst : QList<Point3d> *, Ne : int) : void
+Treshold(dst1 : QList<Point3d> *, aux : QList<float> *, Ne : int) : void
+clustering(ptos : QList< Arbolcluster * > *) : Arbolcluster *
+~Distancia()
Arbol_Cluster
-valor : double
-derecho : Arbol_Cluster
-izquierdo : Arbol_Cluster
-desde : int
-hasta : int
-Buscarcorte(profundidad : int, result : Qlist<int>)
+Arbol_Cluster()
+Arbol_Cluster(valor : double, pos : int)
+get_valor() : double
+get_derecho() : Arbol_Cluster
+get_izquierdo() : Arbol_Cluster
+get_inicio() : int
+get_fin() : int
+listadokeyframe(porciento : int, result : QList<int>)
+profundidad() : int
+es_hoja() : boolean

Figura 8: Diagrama de clases del diseño

Breve descripción de las clases:

I_Plugin_Logica: Clase encargada de identificar cuáles de los *Plugins* de lógica que contiene la plataforma es el que se va a ejecutar según la acción solicitada.

I_Plugin_Ejecucion: Clase encargada de identificar cuáles de los *plugins* de ejecución perteneciente a la lógica indicada es el que se ejecuta.

I_Observador_Ejecucion: Clase encargada de conocer el estado de la ejecución de los *plugins*, para que a su vez la Plataforma conozca datos como: porciento de ejecución, ocurrencia de errores en un proceso de ejecución o la culminación de un proceso en ejecución para que la plataforma lo elimine de la lista de procesos en ejecución.

I_Publicador_ejecucion: Contiene los *plugins* de comunicación que se publican para interactuar con diferentes tecnologías proveniente de sistemas externos.

Distancia: Esta clase es la que se encarga de todo el proceso de cambio de toma.

Extraer_Fotograma: Esta clase es la que se encarga de todas las funcionalidades del *plugin* de lógica.

Extraer_Fotograma: Esta clase es la que se encarga de todas las funcionalidades del *plugin* de ejecución.

Arbol_Cluster: Se encarga de crear un árbol para posteriormente poder realizar el agrupamiento jerárquico.

Hilo_Observer: Esta clase es la que se encarga de mostrar el porciento por donde se va ejecutando la aplicación.

Hilo_Fotograma: Esta clase es la que contiene todo lo relacionado con las operaciones del *plugin*.

Patrones de diseño

Los patrones de diseño son descripciones de clases cuyas instancias colaboran entre sí. Cada patrón es adecuado para ser adaptado a un cierto tipo de problema. Representa un esquema o microarquitectura que supone una solución a problemas (dominios de aplicación) semejantes; una estructura común que tienen aplicaciones semejantes.

A continuación se relacionan los patrones de diseño empleados y su aplicación en la realización de los diagramas de clases del diseño para su posterior implementación.

Los patrones generales de software para asignar responsabilidades (por sus siglas en ingles GRASP) son los que se encargan de asignar las responsabilidades a las clases (Recabarren, 2009).

En esta investigación se han definidos los siguientes patrones GRASP:

Creador: Este patrón como su nombre lo indica es el que crea, el guía la asignación de responsabilidades relacionadas con la creación de objetos, se asigna la responsabilidad de que una clase B cree un Objeto de la clase A.

En la clase de Distancia se pone de manifiesto este patrón debido que se hace necesario crear una instancia de la clase Arbol_Cluster para poder realizar el agrupamiento jerárquico.

Bajo acoplamiento: El acoplamiento es una medida de fuerza con que un elemento está, tiene conocimiento de, confía en, otros elementos. Este patrón es un principio que asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema.

Este patrón se pone de manifiesto en la clase Hilo_fotograma debido a que ella es la clase principal y es la que delega las responsabilidades al *plugin*, diciéndole a la clase Distancia que es la encargada de determinar la distancia entre dos fotogramas consecutivas y así no sobrecargándose a ella misma.

Alta cohesión: La cohesión es una medida de la fuerza con la que se relacionan las clases y el grado de focalización de las responsabilidades de un elemento, Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable, una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo.

Este patrón va estrechamente relacionado con el bajo acoplamiento debido a que cada clase es capaz de realizar sus propias funcionalidades.

2.7 Conclusiones parciales

En el presente capitulo tras la selección de los requisitos funcionales y no funcionales, se ha descrito paso a paso el componente capaz de realizar el resumen de video. Además, siguiendo la metodología *OpenUP*, se ha logrado obtener las características fundamentales que debe poseer el componente a desarrollar. El estudio del algoritmo demuestra que se puede evidenciar todo un proceso con las imágenes, así como poder calcular el histograma de color siendo este un paso fundamental para posteriormente obtener el resultado final. También este algoritmo sirvió de base para poder entender de una mejor manera el proceso para detectar cambios de tomas, basándose en el cálculo de la diferencia entre valores de dos fotogramas consecutivos. La utilización de la arquitectura tuberías y filtros permitió mostrar la estructura que debe tener el componente a implementar.

Capítulo 3: Implementación y prueba

En la realización del presente capítulo se abordan dos fases fundamentales en el ciclo de desarrollo propuesto por la metodología de desarrollo de software OpenUP, las fases de implementación y prueba. Se muestran los principales artefactos que genera y la calidad de lo realizado se muestra mediante las pruebas realizadas como última etapa del desarrollo de la solución.

3.1 Estándares de codificación

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. También se puede entender como un conjunto de reglas de notación y nomenclatura, específicas de cada lenguaje de programación, que se usan y se siguen durante la fase de implementación (codificación) de una aplicación y reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores que no son detectados por lo compiladores, y reducen el tiempo y coste de las actividades de depuración y pruebas necesarias para la detección y corrección de los mismos. La elaboración y el empleo de estándares de codificación ofrecen numerosas ventajas a la hora de elaborar cualquier tipo de producto software y entre sus principales ventajas se pueden mencionar:

- Se reduce la posibilidad de cometer errores.
- Se obtiene un código legible y comprensible.
- Mejora la comunicación entre los miembros del grupo de programadores del equipo de desarrollo.
- Se asegura la legibilidad del código entre distintos programadores, y facilita la compilación del mismo.
- Provee una guía para el encargado de mantenimiento/actualización del sistema con código claro y bien documentado.
- Facilita la portabilidad entre plataformas y aplicaciones.

Para la elaboración de un estándar de codificación se debe tener en cuenta, entre otros aspectos, las especificaciones y características del framework de desarrollo junto al o los lenguajes de programación escogidos. En la presente investigación se propone el uso del siguiente estándar de codificación para seguir con las políticas establecidas por el departamento Señales Digitales:

Tabla 2 Estándares de codificación

Comentarios, separadores, líneas y espacios en blancos		

Ubicación de comentarios.	Inicio de una clase y al final de una línea o bloque de código.	Describir brevemente el objetivo (qué función realiza) de una clase previamente de su declaración se desea especificar la acción específica de una línea de código, utilizar un breve comentario al final de esa línea. Ejemplo:
		//Esto es un comentario al inicio de una clase
		class Distancia:public QObject
		{…}
		QString frame; //este atributo indica una imagen que tiene el archivo de video

Líneas en blanco.	Antes y después de cada función.	Dejar una línea en blanco entre las declaraciones de cada función, se puede utilizar entre líneas cuando el bloque de código es extenso para dar claridad a su entendimiento. Ejemplo: void funcionUno()::otraFuncion() {…} //espacio en blanco void funcionDos()::otraFuncion() {…}
Espacios en blanco.	Entre operadores aritméticos y lógicos.	Usar un espacio en blanco entre los miembros de operaciones aritméticas y lógicas. Ejemplo: media_i = new Distancia(); samplingRate = media_i->AjustarDistancia();
Clases, objetos, funciones, atributos		
Apariencia de clases y funciones.	Se tiene en cuenta las especificaciones del framework.	Se establece usar la notación *Camel Casing* (es una norma de notación que establece que las palabras compuestas debe ser escrita con mayúscula excepto la primera palabra) para nombrar las clases y las funciones. Ejemplo:

		void detecrminar distancia::obtenerDatosVideo() { ... }
Nombre de clases y objetos.	Relacionados con el propósito de su función.	Nombrar las clases de manera tal que con solo leerla se note su objetivo. Ejemplo: class Distancia:public QObject {…}
Apariencia de atributos.	Letras minúsculas.	Los atributos deben ser escritos en minúsculas y su nombre debe guardar relación con el valor que almacena. Ejemplo: QString frame;

3.2 Implementación.

En la implementación se empezó con el resultado del diseño y se implementó el sistema en términos de componentes, es decir, ficheros de código fuente, ejecutables y similares. Los propósitos de la implementación son: (James, 2005).

- Planificar las integraciones de sistema necesarias en cada iteración. Para ellos un enfoque incremental, lo que da lugar a un sistema que se implementa en una sucesión de pasos pequeños y manejables.
- Implementar las clases y subsistemas encontrados durante el diseño. En particular, las clases se implementa como componentes de fichero que contienen código fuente.
- Probar los componentes individualmente, y a continuación integrarlos compilándolos y enlazándolos en uno o más ejecutables, antes de ser enviados para ser integrados y llevar a cabo las comprobaciones de sistema.

3.2.1 Bibliotecas utilizadas de QT.

Qt brinda facilidades a los programadores que le confieren una alta calidad. Las bibliotecas de Qt ofrecen clases y propiedades que se pueden utilizar para la implementación de diferentes programas. En esta investigación se utilizaron varias bibliotecas, algunas de estas son:

- **QObject:** es la clase base de todos los objetos de Qt, es el corazón del modelo de objetos de Qt. La característica central de este modelo es un mecanismo muy poderoso para la comunicación sin fisuras objeto denominado señales y slots.

- **QThread:** proporciona hilos independientes del sistema, comparte datos con todos los otros hilos dentro del proceso, pero se ejecuta de forma independiente en la forma en que un programa separado hace en un sistema operativo multitarea. En lugar de comenzaren main (), QThreads comienza a ejecutarse en el run ().
- **QString:** proporciona una cadena de caracteres Unicode. QString almacena una serie de QCharsde 16 bits, donde a cada QChar le corresponde un Unicode de 4.0 caracteres. Unicode es un estándar internacional que soporta la mayoría de los sistemas de escritura en uso hoy en día. Es un súper-conjunto de US-ASCII (ANSI X3.4-1986) y Latin-1 (ISO 8859-1), y todos los personajes US-ASCII/Latin-1 están disponibles en las mismas posiciones del código.
- **QFile:** proporciona una interfaz para leer y escribir a los archivos, es un dispositivo de entrada y salida para leer y escribir archivos de texto, binarios y recursos, puede ser utilizado por sí mismo o con un QTextStream o QDataStream.
- **QTime:** proporciona funciones de reloj de tiempo, contiene un reloj de tiempo. Se puede leer la hora actual del reloj del sistema y medir un lapso de tiempo transcurrido. Se proporciona funciones para compararlos tiempos y para la manipulación de un tiempo mediante la adición de un número de milisegundos.
- **QDebug:** proporciona un flujo de salida para la información de depuración, se utiliza cada vez que el desarrollador tiene que escribir la depuración o la información de seguimiento a un dispositivo, archivo, cadena o una consola.

3.2.2 Algunas funciones utilizada de la biblioteca OpenCV

- **cv::CvNamedWindow:** Permite crear una ventana.
- **cv::CvtColor:** Permite convertir las imágenes a otro espacio de color.
- **cv::CalcHist:** Calcula la distancia de los histogramas de dos imágenes.
- **cv::CompareHist:** Compara 2 histogramas.
- **cv::MeanStdDev:** Calcula la media y la desviación de una imagen.
- **cv::VideoCapture:** Clase que permite capturar los archivos de video de una cámara.
- **cv::CV_CAP_PROP_FRAME_COUNT**: Permite obtener la cantidad de fotogramas de un video.
- **cv::Release:** Permite liberar o descargar las imágenes.

- **cv::CV_CAP_PROP_FPS:** Permite obtener la cantidad de fotogramas por segundo de un video.
- **cv::Imwrite:** Permite guardar o salvar una imagen.

3.3 Prueba de software

Las pruebas tienen gran importancia en el desarrollo de un software, ya que mediante éstas se pueden detectar y corregir errores tempranamente. Antes de entregar el producto al cliente final se debe garantizar que el software cumpla con todos los requerimientos y se han corregido todos los errores, pues cada vez que el programa se ejecuta, el cliente lo está probando, por tanto, debemos hacer un intento especial para que se sienta satisfecho. Con el objetivo de encontrar el mayor número posible de errores, las pruebas deben conducirse sistemáticamente y los casos de prueba deben diseñarse utilizando técnicas definidas.Un caso de prueba especifica cómo probar un Caso de Uso o un escenario específico del mismo.

Tipos de pruebas:

Pruebas de rendimiento: Son las pruebas que se realizan, desde una perspectiva, para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo.

Caja blanca: Las pruebas de la caja blanca se realizan sobre las funciones internas de un módulo en concreto. Entre las técnicas usadas se encuentran; la cobertura de caminos, pruebas sobre las expresiones lógico-aritméticas, pruebas de camino de datos y comprobación de bucles.

Integración: Es la prueba de la integración de varios componentes que han sido probados como unidades independientes. La integración es costosa y el costo proviene de la prueba.

Pruebas de caja blanca: camino básico

Es una técnica de caja blanca que permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución (JOBP, 2013).

Para aplicar la técnica del camino básico se debe introducir la notación para la representación del flujo de control, este puede representarse por un Grafo de Flujo en el cual:

- Cada nodo del grafo corresponde a una o más sentencias de código fuente.
- Todo segmento de código de cualquier programa se puede traducir a un Grafo de Flujo.
- Se calcula la complejidad ciclomática del grafo.

Un grafo de flujo está formado por 3 componentes fundamentales que ayudan a su elaboración y comprensión, estos brindan información para confirmar que el trabajo se está haciendo adecuadamente.

Componentes del grafo de flujo:

Nodo: son los círculos representados en el grafo de flujo, el cual representa una o más secuencias del procedimiento, donde un nodo corresponde a una secuencia de procesos o a una sentencia de decisión. Los nodos que no están asociados se utilizan al inicio y final del grafo.

Aristas: son constituidas por las flechas del grafo, son iguales a las representadas en un diagrama de flujo y constituyen el flujo de control del procedimiento. Las aristas terminan en un nodo, aun cuando el nodo no representa la sentencia de un procedimiento.

Regiones: son las áreas delimitadas por las aristas y nodos donde se incluye el área exterior del grafo, como una región más. Las regiones se enumeran siendo la cantidad de regiones equivalente a la cantidad de caminos independientes del conjunto básico de un procedimiento.

A continuación se muestra la prueba de caja blanca, utilizando el método del camino básico, al método Clustering:

La complejidad ciclomática está dada por la siguiente formula:

$$CC = CA - NN + 2$$

Dónde:

CC: es la complejidad ciclomática del grafo.

CA: es la cantidad de aristas del grafo.

NN: es la cantidad de nodos del grafo.

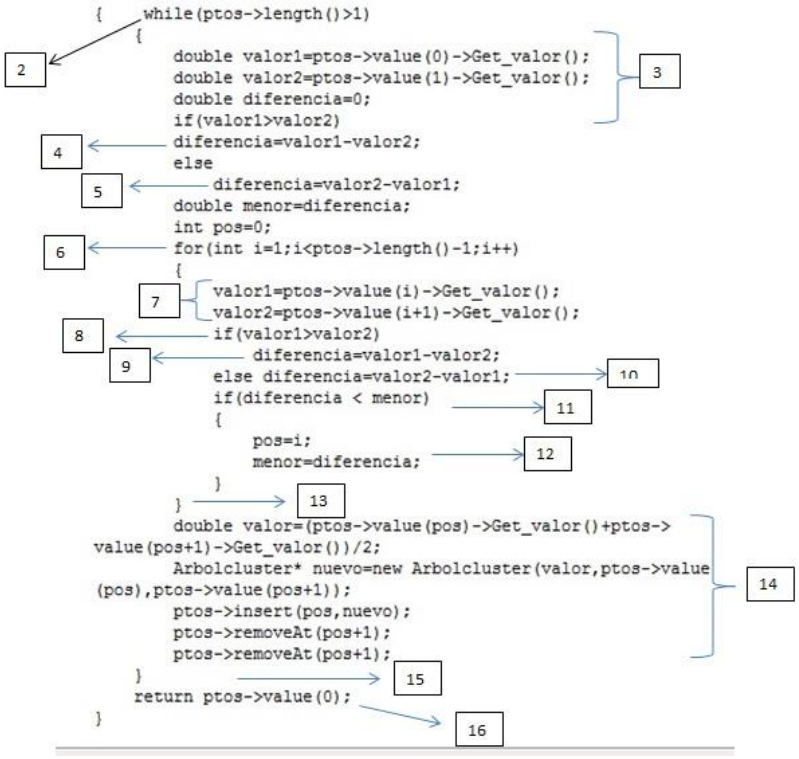

Figura 9 : Método clustering

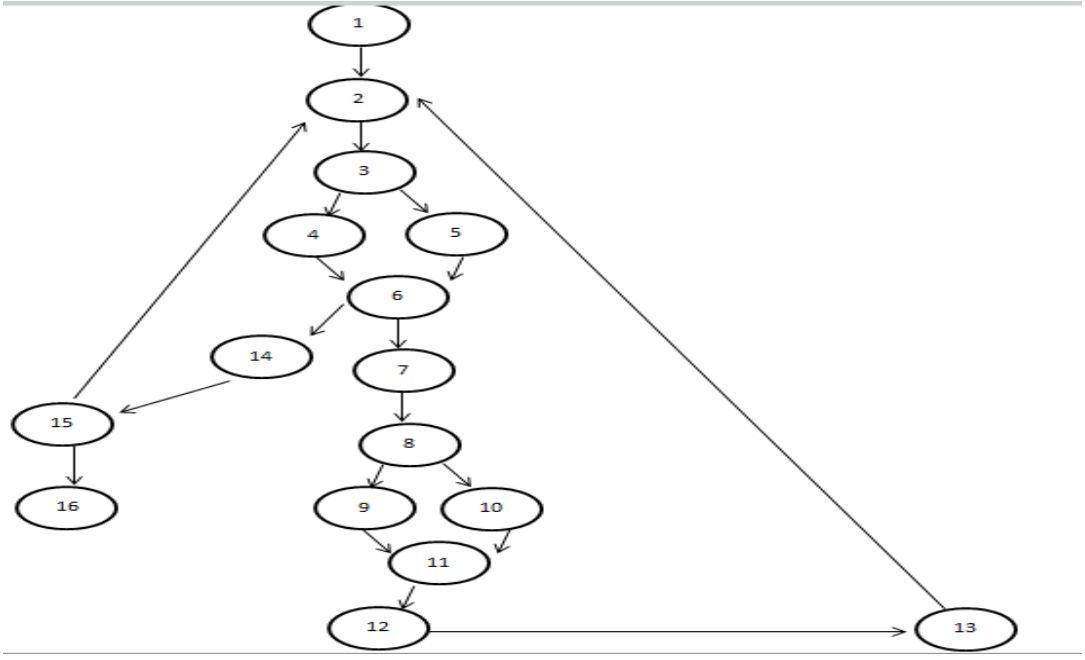

Figura 10: Prueba de caja blanca al método clustering

CC= CC=CA-NN+2

CC= 19-16+2

CC=5

Camino1: 1-2-3-4-6-7-8-9-11-12-13-2-3-4-6-14-15-16

Camino2: 1-2-3-5-6-7-8-10-11-12-13-2-3-5-6-14-15-16

Camino3: 1-2-3-5-6-7-8-9-11-12-13-2-3-4-6-14-15-16

Camino4: 1-2-3-4-6-7-8-9-11-12-13-2-3-4-14-15-2-3-4-6-14-15-16

Camino5: 1-2-3-5-6-7-8-10-11-12-13-2-3-5-6-14-15-2-3-4-6-14-15-16

Resultados: Se detectó una no conformidad, la cual ya fue resuelta en su totalidad.

Pruebas de Rendimiento:

Estas pruebas se sustentan en la necesidad de minimizar las falsas detecciones durante el proceso de segmentación de video para su representación sintética. La forma más utilizada para medir el buen funcionamiento de los algoritmos de detección de cambios de tomas consiste en calcular su *recall (R)* y su *precisión (P).* Para probar los algoritmos con materiales audiovisuales su definición es la siguiente:

$$\boldsymbol{R} = \frac{correctos}{correctos + falsos\ negativos}$$

$$\boldsymbol{P} = \frac{correctos}{correctos + falsos\ positivos}$$

Una medida que combina $Precisión$ y $Recall$ es F que mientras mayor sea su valor, mejor se considera el rendimiento del sistema.

$$F = 2 * \frac{P * R}{P + R}$$

Una vez obtenida F se puede calcular el porciento de error E que tiene el sistema. (Makhoul, y otros)

$$E = 1 - F$$

Estas pruebas se realizaron macheando el valor de la sensibilidad.

Para su documentación se tuvo en cuenta: nombre de la media que se analiza (NM), cantidad de fotogramas (CF), cantidad de cambios abruptos (CDA), falsos positivos (FP), falsos negativos(FN), recall y precisión de los algoritmos.

Tras las corridas del algoritmo, con una sensibilidad=100, se obtuvieron los siguientes resultados.

Tabla 3 Prueba de rendimiento con 100 % de sensibilidad

NM	CF	CDA	C	FP	FN	R	P	F
Aventura.mpg	3950	26	24	3	1	96.0	88.8	92.45
3MSC.mpg	3853	95	91	5	3	96.8	94.7	95.73
Palante.avi	8246	201	189	16	2	98.9	92.1	95.37
Aveces.avi	3925	51	51	5	0	100.0	91.0	95.28

Tabla 4 Prueba de rendimiento con 70 % de sensibilidad

NM	CF	CDA	C	FP	FN	R	P	F
Aventura.mpg	3950	26	24	3	1	96.0	88.8	87.52
3MSC.mpg	3853	95	91	4	2	97.8	94.7	96.22
Palante.avi	8246	201	189	15	2	98.5	92.6	95.45
Aveces.avi	3925	51	51	4	0	100	92.7	96.21

Tabla 5 Prueba de rendimiento con 50 % de sensibilidad

NM	CF	CDA	C	FP	FN	R	P	F
Aventura.mpg	3950	26	24	2	1	96.0	92.3	94.11
3MSC.mpg	3853	95	92	4	0	100	95.8	97.85
Palante.avi	8246	201	198	14	1	99.4	93.3	95.78
Aveces.avi	3925	51	51	3	0	100	94.0	96.90

Tabla 6 Prueba de rendimiento con 20 % de sensibilidad

NM	CF	CDA	C	FP	FN	R	P	F
Aventura.mpg	3950	26	16	2	8	66.6	88.8	76.11
3MSC.mpg	3853	95	60	1	13	82.1	98.3	89.47
Palante.avi	8246	201	130	1	25	83.8	99.2	90.85
Aveces.avi	3925	51	20	2	7	74.0	90.9	81.58

Analizando los resultados anteriores se puede observar que el algoritmo funciona con un recall por encima de 96, esto se debe a que existe un ínfimo número de falsos negativos, siempre menor que 3 excepto en el caso de que la sensibilidad es igual a 20, lo que implica que con este valor no se obtiene correctamente lo que se espera del algoritmo.

En el caso de la precisión del algoritmo con números superiores a 88, si se analiza de forma general, es menor que su recall, en todos los casos excepto cuando el valor de la sensibilidad es 20, debido a que como menor cantidad de sensibilidad se proporciona la detección de mayores falsos negativos por lo que se recomienda que se trabaje con una sensibilidad mayor o igual que 90 por ciento o lo más cercano a 100 posible, que es donde el algoritmo funciona más estable, o sea demuestra mejor los resultados que se están esperando, debido al uso del agrupamiento jerárquico que permite eliminar un gran número de redundancias existentes en el resumen.

Pruebas de integración:

Se logró realizar la integración del componente correctamente.

3.4 Conclusiones parciales

Luego de la implementación del algoritmo expuesto se demuestra que la biblioteca OpenCV fue de vital apoyo para lograr este proceso de implementación, además se logró obtener el componente descrito en capítulos anteriores. El proceso de integración, ayudó a que todos los usuarios del SGPM pueden utilizar de esta funcionalidad. A través de las pruebas realizadas se demuestra que el componente cumple con las necesidades de resumen de video del departamento Señales Digitales, además las pruebas de caja blanca arrojan como resultado que el algoritmo tiene un correcto funcionamiento interno. Las pruebas de rendimiento demuestran que con una sensibilidad igual a 100% se posee de una eficiencia superior al 94 por ciento por lo que se recomienda que se utilice este valor para poder obtener un mejor resumen de video.

Conclusiones generales

En la presente investigación se cumplieron satisfactoriamente todos los objetivos trazados, realizando completamente cada una de las tareas propuestas. A continuación se brindan algunos elementos a modo de conclusión:

- La metodología de desarrollo de software permitió generó los artefactos que dieron solución al problema planteado al inicio de la investigación, permitiendo el entendimiento interno del sistema, facilitando su implementación.
- Con el uso de la biblioteca OpenCV, se logró, un componente capaz de procesar imágenes, para así poder lograr un resumen de video en forma de *storyboard*.
- El algoritmo desarrollado estuvo enfocado en la determinación de resumen de video, satisfaciendo las necesidades de los sistemas que se integran al SGPM y necesitan de esta funcionalidad.
- El método de clasificación del umbral, con su sencillez y su efectividad el cual se complementó con el agrupamiento jerárquico logro detectar los cambios de toma del video.
- Las pruebas realizadas demuestran que el algoritmo como mejor funciona es con un valor de sensibilidad lo más cercano a 100 posible (90-99), por lo que se recomienda usar, los mismos.
- Luego de haber obtenido el *plugin*, se logró mejorar la información visual en la representación sintética de videos en el SGPM.

Recomendaciones

De acuerdo al trabajo realizado, la importancia conferida a la realización de este componente, y teniendo en cuenta los resultados o beneficios que proporciona esta investigación, se recomienda:

- Continuar con el desarrollo de esta investigación con el fin de incorporar elementos que permitan la detección de cambio de tomas graduales en videos.
- Explotar al máximo la información semántica de los videos para realizar resúmenes visuales.

Bibliografía referenciada

GSMspain. 1996. GSMspain. [En línea] GSMspain S.A, 1996. [Citado el: 4 de Diciembre de 2012.] http://www.gsmspain.com/glosario/?palabra=P%CDXEL.

Adrian Kaehler, Gary Bradski. 2008. *Leraning OpenCV.* 2008.

Albiol Colomer, Antonio. 2003. *Seguimiento de objetos en secuencias de video.* Valencia : s.n., 2003.

Alfaro; Félix Murillo. 1999. *Herramientas CASE. Instituto Nacional de Estadísticas e Informática.* [En línea] 11 de 1999. [Citado el: 10 de Diciembre de 2012.] http://www.inei.gob.pe. 875-99-OI-OTDETI-INE.

Andrés, Ing. Rolando Rodríguez. 2008. Herramientas para el modelado y análisis de procesos. [En línea] 2008.

Aventura Technologies, Inc. Aventura. *Aventura. Soluciones de Seguridad de Diseño.* [En línea] Aventura Technologies, Inc. [Citado el: 21 de 12 de 2012.] http://www.aventuracctv.com/es/.

Bass, Leonardo. 2008. *Software Architecture in Practice.* 2008.

Benini, S. 2006. *Extraction of significant video summaries by dendrogram analysis.* 2006.

Bescòs Cisneros, Jesus, y otros. 2006. *IEEE TRANSACTIONS ON MULTIMEDIA, A Unified Model for Techniques on Video-Shot Transition Detection.* 2006.

Blanchete, Jasmin y Summerfield, Mark. 2008. *C++ GUI Programming with Qt 4.* 2008. 978-0132354165.

Booch-Jacobson-Rumbaugh. 2005. *El lenguaje unificado de modelado, manual de referencia.* 2005.

C. Gonzalez, Rafael y E. Woods, Richard. 2002. *Digital Image Processing.* Upper Saddle River, New Jersey : Prentice-Hall, Inc, 2002. 0-201-18075-8.

Chacón, Julio Cesar Rueda. 2006. Aplicación de la Metodología RUP para el desarrollo. Guatemala : s.n., 2006.

Chen, Y. 2010. *A temporal video segmentation and summary generation method based on shot's brup and gradual transition boundary detecting.* 2010.

Cinegy LLC. 2012. [En línea] 2012. [Citado el: 12 de Diciembre de 2012.] http://www.cinegy.com/jml/index.php/en/products-mainmenu-50/mam-menu/workspace-menu.html.

Cisco. 2011. Cisco.com. [En línea] 2011. [Citado el: 21 de Diciembre de 2012.] www.cisco.com .

Diagrama de Secuencia. [En línea] [Citado el: 12 de 4 de 2013.] http://kovachi.sel.inf.uc3m.es/@api/deki/files/81/=Diagramas_de_Secuencia.pdf.

Duck, Black. 2010. Ohloh. *Ohloh BY BLACK DUCK.* [En línea] Black Duck Software, Inc., 2010. [Citado el: 26 de 1 de 2013.] http://www.ohloh.net/p/ffmpeg.

EE Times University. 2012. EE Times. *EE Times Design.* [En línea] EE Times University, 10 de Octubre de 2012. [Citado el: 21 de 12 de 2012.] http://www.eetimes.com/.

FFMPEG. *FFMPEG.* [En línea] [Citado el: 26 de Enero de 3013.] http://www.ffmpeg.org/about.html.

G4S Technology Ltd. 2012. G4S Securing Your World. *G4S Securing Your World. Video Content Analytics.* [En línea] G4S Technology Ltd, 2012. [Citado el: 21 de Diciembre de 2012.] http://www.g4stechnology.co.uk/.

Giraldo, Luis y Zapata, Yuliana. 2005. *Herramientas de Desarrollo de Ingenieria de SW para Linux.* 2005.

González, Danays Rodríguez Martínez, Marilys Valiente. 2010. *Módulo para la gestión de configuración del equipamiento tecnológico en la Universidad de las Ciencias Informáticas.* Habana : s.n., 2010.

González, San Ignacio. 2011. *Definición de video.* 2011.

GSMspain. GSMspain. [En línea] GSMspain. [Citado el: 19 de diciembre de 2012.] http://www.gsmspain.com/glosario/?palabra=P%CDXEL.

http://definicion.de/video/. 2008. Definicion.de. . [En línea] 2008. [Citado el: 10 de Diciembre de 2012.] http://definicion.de/video/..

Informática, Historia de la. 2008-2009. *Metodologías de desarrollo de software.* La Habana : s.n., 2008-2009.

Ingenieria del Software. Séptima edición. **Sommerville, Ian. 2005.** 84-7829-074-5, Madrid : Pearson Educación, 2005. ISBN.

Instituto Nacional de Tecnologías Educativas y de formación de Profesorado. Cultura audiovisual: INTEF. *Sitio Web de INTEF.* [En línea] [Citado el: 12 de noviembre de 2012.] http://recursostic.educacion.es/artes/plastic/web/cms/index.php?id=579.

Jacobson James. 2000. *El lenguaje Unificado de Modelado, Manual de referencia.* 2000.

—. 2000. *El lenguaje Unificado de Modelado, Manual de referencia.* 2000.

JACOBSON, Ivar y BOOCH, Grady. 2000. *El Proceso Unificado de Desarrollo de Software.* 2000.

James, Jacobson. 1999. *El proceso unificado de desarrollo de software.* Madrid : s.n., 1999.

Jan Di, Chong. 2005. *Video Summarization and Scene Detection by Graph Modeling.* 2005.

Jimenez, Zhugo. 2004. *Exploring video content structure for hierarchical summarization.* 2004.

JOBP. 2013. *Ingenieria de Software II.* 2013.

KIOSKEA. 2012. KIOSKEA.net. [En línea] 2012. [Citado el: 2012 de 12 de 15.] http://es.kioskea.net/faq/2635-que-es-un-plugin.

La Inteligencia en aplicaciones de video. **Revista de Negocios de Seguridad. 2010.** s.l. : Revista de Negocios de Seguridad, 2010.

Laganière, Robert. 2011. *OpenCV 2 Computer Vision Application Programming .* s.l. : Packt Publishing Ltd., 2011. 978-1-849513-24-1.

Larman, Craig. 2003. *UML y patrones.* 2003.

LLC., Cinegy. Cinegy. [En línea] [Citado el: 12 de Diciembre de 2012.] http://www.cinegy.com/jml/index.php/en/products-mainmenu-50/mam-menu/workspace-menu.html..

Loy, Chen Change. 2010. *Activity Understanding and Unusual Event.* Londres : s.n., 2010.

Lucas HB, Zhang y YJ and Yao, YR. 1999. *Robust Gradual Scene Change Detection.* Beijing : Department of Electronic Engineering : s.n., 1999. 100084..

Marcos, Mari Carmen. 2004. *Tarsys, un software para la gestión de documentos audiovisuales. .* 2004.

Medrano, Carlos. 2009. *El Storyboards, guiones para medios audiovisuales.* 2009.

Molina, R. 1998. *Introducción al Procesamiento y Análisis de Imágenes Digitales.* Granada : s.n., 1998. 18071.

Muñoz, León Alberto Martínez. 2011. *Implementación de un editor gráfico de circuitos.* Colombia : Universidad Politécnica de Cartagena, 2011.

Orallo, Enrique Hernández. 2010. *El Lenguaje Unificado de Modelado (UML).* 2010.

—. 2010. *El Lenguaje Unificado de Modelado (UML).* 2010.

Paul Bustamante, Iker Aguinaga, Miguel Aybar, Luis Olaizola, Iñigo Lazacano. 2004. *Aprenda C++ básico.* navarra : s.n., 2004.

Permuy, Fernando Bellas. 2000. *El lenguage de programación C++.* Universidad de Coruña : s.n., 2000.

Pixelco. 2012. Pixelco Blog. *Pixelco Media.* [En línea] 2012. [Citado el: 12 de diciembre de 2012.] http://pixelcoblog.com/qt-creator-completo-entorno-de-desarrollo-multiplataforma/.

Pressman, Roger. 2002. *Ingeniería del Software: Un enfoque práctico.* España : McGraw-Hill Companies, 2002. 5ta Edición.

Pressman, Roger S. 2001. *Ingenieria del Software. Un enfoque práctico.* Madrid : s.n., 2001.

Ramiro, Daniel Isaac Khan. 2011. *Interfaz gráfica multiplataforma para la simulación de ecuaciones físicas.* Madrid: Universidad Rey Juan Carlos : s.n., 2011.

Raúl Igual, Carlos Medrano. 2010. *Tutorial de OpenCV.* 2010.

Recabarren, Matìas. 2009. *Patrones Grasp I.* Chile : s.n., 2009.

Richard Bowden, Pakorn KaewTraKulPong. 2011. *An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection.* Brunel : Kluwer Academic Publishers, 2011.

Robert Laganière. 2011. *OpenCV 2 Computer Vision .* Olton : Packt Publishing Ltd, 2011. 1180511.

Roger. 2007. Pressman. Ingieneria del software. *Capítulo 8. Modelado de análisis.* 2007.

Saez, Edmundo. 2006. *Segmentación Automática de Video.* Málaga : Universidad : s.n., 2006.

SAĞLAM, ALĠ. 2009. *ADAPTIVE CAMERA TAMPER DETECTION FOR VIDEO SURVEILLANCE.* 2009.

Sierra, María. 2006. *Trabajando con Visual Paradigm for UML.* Cantabria : s.n., 2006.

2012. Sitio oficial de Opencv. [En línea] 2012. [Citado el: 8 de enero de 2013.] http://opencv.org/.

Smeaton, Alan F., Over, Paul and Doherty y R, Aiden. 2010. *Video shot boundary detection: Seven years of TRECVid activity. s.l. : Computer Vision and Image Understanding.* 2010.

Software Engeneering Lab. 2013. Diagrama de Secuencia. [En línea] 2013. [Citado el: 12 de 4 de 2013.] http://kovachi.sel.inf.uc3m.es/@api/deki/files/81/=Diagramas_de_Secuencia.pdf.

Sommerville, Ian. 2007. *Ingeniería de Software 8va edición.* 2007.

Tarsys, un software para la gestión de documentos audiovisuales. **Marcos, Mari Carmen. 2004.** 2004.

Telestream. 2012. telestream. [En línea] 2012. [Citado el: 12 de Diciembre de 2012.] http://www.telestream.net/telestream-solutions/digital-asset-management.htm.

Tello, Jesus Cáceres. 2012. *Diagramas de Casos de Uso.* 2012.

THRIVE Intelligence, LLC. 2012. THRIVE Intelligence. *THRIVE Intelligence.* [En línea] 2012. [Citado el: 21 de Diciembre de 2012.] www.THRIVEintelligence.com .

Video analytics and preemptive surveillance. **VideoSurveillance.com. 2010.** Portland : VideoSurveillance.com, 2010.

Videoma. 2012. *Videoma: Soluciones .* 2012.

Wah, Chong. 2005. *Video Summarization and Scene Detection by Graph Modeling.* 2005.

Wandelmer Pedro, Jose San. 2006. *Arquitectura Paralela para el procesamiento y anàlisis de video digital utilizando MPEG-21 Aplicaciones implantadas.* Madrid: Universidad Politècnica de Madrid : s.n., 2006.

Wandelmer, Jose San Pedro. 2006. *Arquitectura Paralela para el procesamiento y anàlisis de video digital utilizando MPEG-21 Aplicaciones implantadas.* Madrid: Universidad Politècnica de Madrid : s.n., 2006.

Bibliografía consultada

GSMspain. 1996. GSMspain. [En línea] GSMspain S.A, 1996. [Citado el: 4 de Diciembre de 2012.] http://www.gsmspain.com/glosario/?palabra=P%CDXEL.

Adrian Kaehler, Gary Bradski. 2008. *Leraning OpenCV.* 2008.

Albiol Colomer, Antonio. 2003. *Seguimiento de objetos en secuencias de video.* Valencia : s.n., 2003.

Alfaro; Félix Murillo. 1999. *Herramientas CASE. Instituto Nacional de Estadísticas e Informática.* [En línea] 11 de 1999. [Citado el: 10 de Diciembre de 2012.] http://www.inei.gob.pe. 875-99-OI-OTDETI-INE.

Andrés, Ing. Rolando Rodríguez. 2008. Herramientas para el modelado y análisis de procesos. [En línea] 2008.

Aventura Technologies, Inc. Aventura. *Aventura. Soluciones de Seguridad de Diseño.* [En línea] Aventura Technologies, Inc. [Citado el: 21 de 12 de 2012.] http://www.aventuracctv.com/es/.

Bass, Leonardo. 2008. *Software Architecture in Practice.* 2008.

Benini, S. 2006. *Extraction of significant video summaries by dendrogram analysis.* 2006.

Bescòs Cisneros, Jesus, y otros. 2006. *IEEE TRANSACTIONS ON MULTIMEDIA, A Unified Model for Techniques on Video-Shot Transition Detection.* 2006.

Blanchete, Jasmin y Summerfield, Mark. 2008. *C++ GUI Programming with Qt 4.* 2008. 978-0132354165.

Booch-Jacobson-Rumbaugh. 2005. *El lenguaje unificado de modelado, manual de referencia.* 2005.

C. Gonzalez, Rafael y E. Woods, Richard. 2002. *Digital Image Processing.* Upper Saddle River, New Jersey : Prentice-Hall, Inc, 2002. 0-201-18075-8.

Chacón, Julio Cesar Rueda. 2006. Aplicación de la Metodología RUP para el desarrollo. Guatemala : s.n., 2006.

Chen, Y. 2010. *A temporal video segmentation and summary generation method based on shot's brup and gradual transition boundary detecting.* 2010.

Cinegy LLC. 2012. [En línea] 2012. [Citado el: 12 de Diciembre de 2012.] http://www.cinegy.com/jml/index.php/en/products-mainmenu-50/mam-menu/workspace-menu.html.

Cisco. 2011. Cisco.com. [En línea] 2011. [Citado el: 21 de Diciembre de 2012.] www.cisco.com .

Diagrama de Secuencia. [En línea] [Citado el: 12 de 4 de 2013.] http://kovachi.sel.inf.uc3m.es/@api/deki/files/81/=Diagramas_de_Secuencia.pdf.

Duck, Black. 2010. Ohloh. *Ohloh BY BLACK DUCK.* [En línea] Black Duck Software, Inc., 2010. [Citado el: 26 de 1 de 2013.] http://www.ohloh.net/p/ffmpeg.

EE Times University. 2012. EE Times. *EE Times Design.* [En línea] EE Times University, 10 de Octubre de 2012. [Citado el: 21 de 12 de 2012.] http://www.eetimes.com/.

FFMPEG. *FFMPEG.* [En línea] [Citado el: 26 de Enero de 3013.] http://www.ffmpeg.org/about.html.

G4S Technology Ltd. 2012. G4S Securing Your World. *G4S Securing Your World. Video Content Analytics.* [En línea] G4S Technology Ltd, 2012. [Citado el: 21 de Diciembre de 2012.] http://www.g4stechnology.co.uk/.

Giraldo, Luis y Zapata, Yuliana. 2005. *Herramientas de Desarrollo de Ingenieria de SW para Linux.* 2005.

González, Danays Rodríguez Martínez, Marilys Valiente. 2010. *Módulo para la gestión de configuración del equipamiento tecnológico en la Universidad de las Ciencias Informáticas.* Habana : s.n., 2010.

González, San Ignacio. 2011. *Definición de video.* 2011.

GSMspain. GSMspain. [En línea] GSMspain. [Citado el: 19 de diciembre de 2012.] http://www.gsmspain.com/glosario/?palabra=P%CDXEL.

http://definicion.de/video/. 2008. Definicion.de. . [En línea] 2008. [Citado el: 10 de Diciembre de 2012.] http://definicion.de/video/..

Informática, Historia de la. 2008-2009. *Metodologías de desarrollo de software.* La Habana : s.n., 2008-2009.

Ingenieria del Software. Séptima edición. **Sommerville, Ian. 2005.** 84-7829-074-5, Madrid : Pearson Educación, 2005. ISBN.

Instituto Nacional de Tecnologías Educativas y de formación de Profesorado. Cultura audiovisual: INTEF. *Sitio Web de INTEF.* [En línea] [Citado el: 12 de noviembre de 2012.] http://recursostic.educacion.es/artes/plastic/web/cms/index.php?id=579.

Jacobson James. 2000. *El lenguaje Unificado de Modelado, Manual de referencia.* 2000.

—. 2000. *El lenguaje Unificado de Modelado, Manual de referencia.* 2000.

JACOBSON, Ivar y BOOCH, Grady. 2000. *El Proceso Unificado de Desarrollo de Software.* 2000.

James, Jacobson. 1999. *El proceso unificado de desarrollo de software.* Madrid : s.n., 1999.

Jan Di, Chong. 2005. *Video Summarization and Scene Detection by Graph Modeling.* 2005.

Jimenez, Zhugo. 2004. *Exploring video content structure for hierarchical summarization.* 2004.

JOBP. 2013. *Ingenieria de Software II.* 2013.

KIOSKEA. 2012. KIOSKEA.net. [En línea] 2012. [Citado el: 2012 de 12 de 15.] http://es.kioskea.net/faq/2635-que-es-un-plugin.

La Inteligencia en aplicaciones de video. **Revista de Negocios de Seguridad. 2010.** s.l. : Revista de Negocios de Seguridad, 2010.

Laganière, Robert. 2011. *OpenCV 2 Computer Vision Application Programming .* s.l. : Packt Publishing Ltd., 2011. 978-1-849513-24-1.

Larman, Craig. 2003. *UML y patrones.* 2003.

LLC., Cinegy. Cinegy. [En línea] [Citado el: 12 de Diciembre de 2012.] http://www.cinegy.com/jml/index.php/en/products-mainmenu-50/mam-menu/workspace-menu.html..

Loy, Chen Change. 2010. *Activity Understanding and Unusual Event.* Londres : s.n., 2010.

Lucas HB, Zhang y YJ and Yao, YR. 1999. *Robust Gradual Scene Change Detection.* Beijing : Department of Electronic Engineering : s.n., 1999. 100084..

Marcos, Mari Carmen. 2004. *Tarsys, un software para la gestión de documentos audiovisuales. .* 2004.

Medrano, Carlos. 2009. *El Storyboards, guiones para medios audiovisuales.* 2009.

Molina, R. 1998. *Introducción al Procesamiento y Análisis de Imágenes Digitales.* Granada : s.n., 1998. 18071.

Muñoz, León Alberto Martínez. 2011. *Implementación de un editor gráfico de circuitos.* Colombia : Universidad Politécnica de Cartagena, 2011.

Orallo, Enrique Hernández. 2010. *El Lenguaje Unificado de Modelado (UML).* 2010.

—. 2010. *El Lenguaje Unificado de Modelado (UML).* 2010.

Paul Bustamante, Iker Aguinaga, Miguel Aybar, Luis Olaizola, Iñigo Lazacano. 2004. *Aprenda C++ básico.* navarra : s.n., 2004.

Permuy, Fernando Bellas. 2000. *El lenguage de programación C++.* Universidad de Coruña : s.n., 2000.

Pixelco. 2012. Pixelco Blog. *Pixelco Media.* [En línea] 2012. [Citado el: 12 de diciembre de 2012.] http://pixelcoblog.com/qt-creator-completo-entorno-de-desarrollo-multiplataforma/.

Pressman, Roger. 2002. *Ingeniería del Software: Un enfoque práctico.* España : McGraw-Hill Companies, 2002. 5ta Edición.

Pressman, Roger S. 2001. *Ingenieria del Software. Un enfoque práctico.* Madrid : s.n., 2001.

Ramiro, Daniel Isaac Khan. 2011. *Interfaz gráfica multiplataforma para la simulación de ecuaciones físicas.* Madrid: Universidad Rey Juan Carlos : s.n., 2011.

Raúl Igual, Carlos Medrano. 2010. *Tutorial de OpenCV.* 2010.

Recabarren, Matìas. 2009. *Patrones Grasp I.* Chile : s.n., 2009.

Richard Bowden, Pakorn KaewTraKulPong. 2011. *An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection.* Brunel : Kluwer Academic Publishers, 2011.

Robert Laganière. 2011. *OpenCV 2 Computer Vision .* Olton : Packt Publishing Ltd, 2011. 1180511.

Roger. 2007. Pressman. Ingieneria del software. *Capítulo 8. Modelado de análisis.* 2007.

Saez, Edmundo. 2006. *Segmentación Automática de Video.* Málaga : Universidad : s.n., 2006.

SAĞLAM, ALĠ. 2009. *ADAPTIVE CAMERA TAMPER DETECTION FOR VIDEO SURVEILLANCE.* 2009.

Sierra, María. 2006. *Trabajando con Visual Paradigm for UML.* Cantabria : s.n., 2006.

2012. Sitio oficial de Opencv. [En línea] 2012. [Citado el: 8 de enero de 2013.] http://opencv.org/.

Smeaton, Alan F., Over, Paul and Doherty y R, Aiden. 2010. *Video shot boundary detection: Seven years of TRECVid activity. s.l. : Computer Vision and Image Understanding.* 2010.

Software Engeneering Lab. 2013. Diagrama de Secuencia. [En línea] 2013. [Citado el: 12 de 4 de 2013.] http://kovachi.sel.inf.uc3m.es/@api/deki/files/81/=Diagramas_de_Secuencia.pdf.

Sommerville, Ian. 2007. *Ingeniería de Software 8va edición.* 2007.

Tarsys, un software para la gestión de documentos audiovisuales. **Marcos, Mari Carmen. 2004.** 2004.

Telestream. 2012. telestream. [En línea] 2012. [Citado el: 12 de Diciembre de 2012.] http://www.telestream.net/telestream-solutions/digital-asset-management.htm.

Tello, Jesus Cáceres. 2012. *Diagramas de Casos de Uso.* 2012.

THRIVE Intelligence, LLC. 2012. THRIVE Intelligence. *THRIVE Intelligence.* [En línea] 2012. [Citado el: 21 de Diciembre de 2012.] www.THRIVEintelligence.com .

Video analytics and preemptive surveillance. **VideoSurveillance.com. 2010.** Portland : VideoSurveillance.com, 2010.

Videoma. 2012. *Videoma: Soluciones .* 2012.

Wah, Chong. 2005. *Video Summarization and Scene Detection by Graph Modeling.* 2005.

Wandelmer Pedro, Jose San. 2006. *Arquitectura Paralela para el procesamiento y anàlisis de video digital utilizando MPEG-21 Aplicaciones implantadas.* Madrid: Universidad Politècnica de Madrid : s.n., 2006.

Wandelmer, Jose San Pedro. 2006. *Arquitectura Paralela para el procesamiento y anàlisis de video digital utilizando MPEG-21 Aplicaciones implantadas.* Madrid: Universidad Politècnica de Madrid : s.n., 2006.

Glosario de términos

Archivo: Espacio que se reserva en el dispositivo de memoria de un computador para almacenar porciones de información que tienen la misma estructura y que pueden manejarse mediante una instrucción única.

Dendograma: representación gráfica en forma de árbol que ilustra el proceso de agrupación en un análisis de clúster.

Escalabilidad: la medida de la capacidad de crecimiento de un servicio o de una aplicación para satisfacer demandas de rendimiento cada vez mayores.

HSV: el modelo HSV (del *inglés Hue, Saturation, Value* – Matiz, Saturación, Valor), también llamado HSB (*Hue, Saturation, Brightness* – Matiz, Saturación, Brillo), define un modelo de color en términos de sus componentes.

Multimedia: archivo que utiliza conjunta y simultáneamente diversos medios, como imágenes, sonidos y texto, en la transmisión de una información.

Píxel: superficie homogénea más pequeña de las que componen una imagen, que se define por su brillo y color.

Vector: un conjunto de variables del mismo tipo cuyo acceso se realiza por índices.