

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 6



Plugin para la Identificación Automática de Televisoras

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

AUTOR: Ulises Araujo Cohen

TUTOR: Ing. Jean Michael Suárez Pérez

Ciudad de La Habana, junio ___ de 2013.

Año 55 de la Revolución

DECLARACIÓN DE AUTORÍA

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

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

Ulises Araujo Cohen

Firma del Autor

Jean Michael Suárez Pérez

Firma del Tutor

DATOS DE CONTACTO

Tutor:

Nombre y apellidos: Jean Michael Suárez Pérez.

Sexo: M F

Institución: Universidad de las Ciencias Informáticas.

Correo electrónico: jmsuarez@uci.cu.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2009.

Institución donde se graduó: Universidad de las Ciencias Informáticas.

Labora en el proyecto Sistema Gestor de Procesos de Media del departamento Señales Digitales de la Universidad de las Ciencias Informática. En la producción cumple los roles de desarrollador y jefe de proyecto. Está categorizado de instructor y ha colaborado con publicaciones internacionales.

DEDICATORIA

A mi mamá.

AGRADECIMIENTOS

A mis padres, motores incansables de mi superación, por su eterna confianza e insustituible guía.

A la familia por el apoyo en los momentos más difíciles.

A Rubén, Jessica, Alfredo y Daniel por tanto amor y paciencia.

A Héctor su incondicionalidad.

A Ernesto por su sacrificio, sin él nada de esto sería posible.

A Kenia, Wendy y Anabel, por enseñarme tanto en tan poco tiempo.

A Cristóbal y Yanelis protagonistas silenciosos de mi andar universitario.

A los que se fueron, de los que también aprendí.

A Jean Michael, Sisley, Reinel, Mariemy, Miriam, Andris, Pupo, Pepo, Aindamais, David y tantos que de una forma u otra siempre creyeron en mí.

A todos mis profesores, en especial los más exigentes, que resultaron ser los mejores.

A todos, mi eterna gratitud...

Resumen:

Los productos realizados por el departamento Señales Digitales tienen una fuerte competencia en el mercado internacional, por esta razón se ha buscado incorporarles funcionalidades útiles y atractivas teniendo en cuenta potenciales clientes. Para dar respuesta a dichas necesidades y a las dificultades detectadas en el proceso de identificación de televisoras se elaboró un plugin para automatizar esta tarea. El mismo se integra a la Plataforma de Codificación e Indexación, subsistema del Sistema Gestor de Procesos de Media perteneciente al departamento.

Durante los tres capítulos que conforman este documento se describió el desarrollo de un plugin capaz de identificar de forma automática las televisoras presentes en un material audiovisual. Se representaron las distintas etapas de su desarrollo: fundamentación teórica, implementación y validación; demostrándose la factibilidad de su integración a la Plataforma de Codificación e Indexación. Dentro de los resultados de la investigación realizada se tiene toda la documentación establecida en el proceso de desarrollo del plugin así como sus entregables en software.

Palabras claves:

Logo, Monitorización, Patrones, Plugin, Televisoras.

Índice

Índice	VI
Índice de Figuras	IX
Índice de Tablas	X
Introducción	1
CAPÍTULO 1: Fundamentación Teórica	5
1.1 Introducción	5
1.2 Conceptos asociados al dominio del problema	5
1.2.1 Imagen digital.....	5
1.2.2 Video digital.....	5
1.2.3 Marcas de agua	6
1.2.4 Patrón	6
1.2.5 Logo.....	7
1.2.6 Televisora.....	7
1.3 Objeto de Estudio.	7
1.3.1 Descripción General.....	7
1.3.2 Marcas de agua	8
1.3.3 Integrales proyectivas	8
1.3.4 Template Matching.....	9
1.3.5 Descripción actual del dominio del problema	10
1.3.6 Situación problemática.....	10
1.4 Soluciones existentes	11
1.4.1 Teletrax	11
1.4.2 Videoma Broadcast Monitor (VDM).....	12
1.5 Tecnologías y herramientas para el desarrollo del plugin para la identificación automática de televisoras	13
1.5.1 Proceso Unificado de Desarrollo de Software (RUP). Metodología de desarrollo.....	13
1.5.2 Lenguaje Unificado de Modelado (UML).....	14
1.5.3 Herramienta de modelado de software	15

1.5.4	Visual Paradigm 8.0	16
1.5.5	Lenguaje de programación C++	16
1.5.6	Marco de trabajo	17
1.5.7	Qt	17
1.5.8	Entorno de Desarrollo Integrado (IDE). Qt Creator 4.8	18
1.5.9	Biblioteca para el procesamiento de imágenes. OpenCV 2.1	18
1.5.10	Sistema Gestor de Bases de Datos (SGBD). Postgre SQL 9.0	19
1.6	Conclusiones parciales	20
CAPÍTULO 2: Análisis y diseño		21
2.1	Introducción	21
2.2	Modelo del Dominio	21
2.3	Descripción del Modelo del Dominio.....	22
2.4	Descripción de las clases del Modelo del Dominio	22
2.5	Requisitos	23
2.5.1	Requisitos Funcionales	23
2.5.2	Requisitos No Funcionales.....	24
2.6	Algoritmo de funcionamiento para el plugin identificador de televisoras.....	25
2.7	Diagramas de Caso de Uso del Sistema (DCUS).....	25
2.7.1	Definición de los actores	26
2.7.2	Especificación de los Casos de Uso del Sistema (CUS).....	26
2.8	Arquitectura del sistema	31
2.9	Patrones de diseño.....	33
2.9.1	Patrones GRASP (General Responsibility Assignment Software Patterns)	33
2.9.2	Patrones GoF (Gang of Four)	34
2.10	Modelo del Diseño.....	35
2.10.1	Diagrama de Clases del Diseño.....	35
2.10.2	Diagrama de Clases del Diseño para el Plugin para la Identificación de Televisoras 35	
2.11	Diagramas de secuencia	36
2.11.1	Diagrama de secuencia CU Monitorizar Media.....	36
2.11.2	Diagrama de secuencia CU Notificar Evento	36

2.11.3	Diagrama de secuencia CU Notificar Porciento	37
2.11.4	Diagrama de secuencia CU Detener Proceso	37
2.12	Descripción de las clases del diseño.....	37
2.13	Conclusiones parciales.....	41
CAPÍTULO 3: Implementación y Pruebas.....		42
3.1	Introducción	42
3.2	Modelo de implementación.....	42
3.2.1	Diagrama de despliegue	42
3.2.2	Diagrama de componentes	43
3.2.3	Diagrama de componentes de código fuente.....	44
3.3	Pruebas de Software	44
3.3.1	Pruebas de Caja Negra.....	45
3.4	Diseños de casos de pruebas	45
3.4.1	Diseños de pruebas de Caja Negra	45
3.5	Resultados de las pruebas	47
3.6	Conclusiones parciales.....	48
CONCLUSIONES GENERALES		49
REFERENCIAS		51
BIBLIOGRAFÍA.....		54

Índice de Figuras

Figura 1 Modelo del Dominio	21
Figura 2 Diagrama de Casos de Uso del Sistema	26
Figura 3 Estructura del Sistema Gestor de Procesos de Media	32
Figura 4 Diagrama de Clases del Diseño	35
Figura 5 Diagrama de secuencia CU Monitorizar Media.....	36
Figura 6 Diagrama de secuencia CU Notificar Evento.....	36
Figura 7 Diagrama de secuencia CU Notificar Por ciento.....	37
Figura 8 Diagrama de secuencia CU Detener Proceso	37
Figura 9 Diagrama de despliegue.....	43
Figura 10 Diagrama de componentes.....	44

Índice de Tablas

Tabla 1 Clientes internacionales para soluciones existentes.....	12
Tabla 2 Descripción del CUS Monitorizar Media	28
Tabla 3 Descripción del CUS Notificar Evento.....	30
Tabla 4 Descripción del CUS Notificar Por ciento.....	30
Tabla 5 Descripción del CUS Detener Proceso	31
Tabla 6 Clase Controladora Identificar Televisora	38
Tabla 7 Clase Entidad Hilo_Identificar	39
Tabla 8 Clase Entidad Logo.....	39
Tabla 9 Clase I_Plugin_Ejecucion	40
Tabla 10 Clase I_Publicador_Ejecucion	40
Tabla 11 Clase I_Observador_Ejecucion.....	41
Tabla 12 Caso de Prueba-Monitorizar Media	46
Tabla 13 Caso de Prueba-Notificar Por ciento.....	46
Tabla 14 Caso de Prueba-Notificar Evento.....	47
Tabla 15 Caso de Prueba-Detener Proceso	47
Tabla 16 Resultado de la primera iteración de pruebas.....	47
Tabla 17 Resultado de la segunda iteración de pruebas	47
Tabla 18 Resultado de la tercera iteración de pruebas.....	47

Introducción

La televisión como medio de difusión ha evolucionado continuamente a la par de los avances tecnológicos; este desarrollo es visible a través de las miles de transmisiones simultáneas que abundan en el espacio radioeléctrico. Debido al nivel de globalización alcanzado por el sector televisivo se ha impuesto la necesidad de crear mecanismos que supervisen las emisiones provenientes del mismo. De esta forma son los gobiernos los principales interesados en utilizar herramientas que permitan mantener un orden en el flujo de la información (auditando horarios, contenido y otras características) sin que se violen los principios legales establecidos para el uso del espacio radioeléctrico de cada nación.

Internacionalmente se utilizan herramientas informáticas que permiten que este control sea efectivo y abarcador, éstas están orientadas a realizar tareas complejas y repetitivas como la monitorización de los usuarios y el registro de su actividad por medio de informes elaborados a partir de metadatos (como la televisora que emite o emitió la señal) o la optimización de los flujos de trabajo en tareas de producción de televisión digital sin cinta. Una de las características que debe poseer cualquiera de estos sistemas es que el contenido se encuentre disponible en el formato adecuado cuando sea necesario y se entregue de forma eficiente, independientemente de la combinación de equipos y sistemas utilizados.

El Departamento Señales Digitales perteneciente al centro Geoinformática y Señales Digitales (GEYSED) de la facultad 6 en la Universidad de las Ciencias Informáticas (UCI) tiene como objetivo equiparar los modelos internacionales a partir del desarrollo de sistemas informáticos que realicen la gestión de materiales audiovisuales. La presente investigación contempla el almacenamiento de metadatos de estos materiales como un proceso vital para el flujo de trabajo de estos sistemas y que en su conjunto permite agilizar posteriores procesos de recuperación. Algunos de estos metadatos identifican la cadena televisiva que está transmitiendo o transmitió el material (lo cual permite realizar filtros de búsquedas para canales televisivos específicos) o los instantes de tiempo en los que comienza o termina determinada transmisión (información útil para herramientas de grabación por eventos).

En las primeras soluciones propuestas por el departamento para almacenamiento y recuperación se registran las apariciones de las televisoras a través de monitorización humana o con la visualización bajo demanda del fichero grabado. Este proceso es agotador puesto que un documentalista debe visualizar todo el material para ver las apariciones o no de espacios referentes a un canal. Una clara representación

de esta labor se da cuando la monitorización es en tiempo real y el usuario de la aplicación debe dar un seguimiento sin pausa a cualquier aparición. Las transmisiones en horas de la madrugada, insuficiente cantidad de auditores, ausencia de registro de aparición por error humano son solo algunos ejemplos de escenarios de riesgo que se presentan en estos casos. La propia naturaleza intensiva de este tipo de control es cara y no siempre fiable. La respuesta en el mundo de la informática para esta situación es la automatización del procedimiento en sí.

La solución que se busca en el departamento no está ajena a las tendencias internacionales, como parte de este existe un proyecto que es el encargado de realizar la automatización de procesos: el Sistema Gestor de Procesos de Media (SGPM) el cuál agrega a su Plataforma de Codificación e Indexación (PCI) operaciones de manera dinámica en forma de plugins y ha sido seleccionado para dar respuesta a la problemática evidenciada en los sistema de recuperación y almacenamiento audiovisual.

Por lo antes expuesto se puede definir el siguiente **problema a resolver**: El registro de identificación de televisoras dentro de los productos del Departamento Señales Digitales es un proceso incompleto e ineficiente al ser realizado de forma manual.

Se define como **objeto de estudio**: Técnicas automáticas para la identificación de televisoras en video. Como **campo de acción**: La identificación automática de canales televisivos en el producto SGPM.

El **objetivo general** es: Desarrollar un plugin para el SGPM que realice el proceso de identificación de canales televisivos en video.

La **idea a defender** propuesta es la incorporación de un plugin en los productos del Departamento Señales Digitales con el fin de automatizar la identificación de televisoras.

Para guiar la presente investigación se definieron las siguientes tareas de la investigación:

- Describir las técnicas automáticas para la identificación de canales televisivos en video.
- Caracterizar métodos o algoritmos desarrollados con anterioridad para la identificación de televisoras y proponer un esquema que se adecue a las necesidades de los productos del Departamento Señales Digitales que lo requieran.
- Identificar requisitos de integración con el SGPM.
- Diseñar el plugin que permita su integración con el SGPM.

- Implementar el plugin para la identificación de canales de televisión en el SGPM.
- Diseñar y aplicar las pruebas al plugin para la identificación de canales de televisión en el SGPM sobre videos heterogéneos.

Para sustentar la idea a defender planteada y en correspondencia con el objetivo y las tareas propuestas se aplicaron los siguientes métodos científicos durante el desarrollo de la investigación:

Métodos Teóricos

Histórico-Lógico:

Permitieron analizar el avance y las tendencias en las aplicaciones y bibliotecas que se dedican a la identificación de canales televisivos en transmisiones audiovisuales.

Analítico-Sintético:

Permitió valorar y definir las distintas técnicas que son posibles emplear para la identificación de canales televisivos y así definir cuál es la que mejor fiabilidad brinda.

Modelación:

Permitió modelar las clases y la creación de artefactos de ingeniería y a continuación desarrollar la aplicación.

Métodos Empíricos

Observación: Se utilizó para el estudio de las tendencias existentes en el medio audiovisual y de esta forma definir patrones de comportamiento para la solución propuesta.

Descripción de los Capítulos

Capítulo 1: En este capítulo se abordan temas referentes a los métodos para la detección de televisoras, sus principales ventajas y desventajas. Se propone la técnica a implementar para la identificación de televisoras en materiales audiovisuales. Además, se comparten conceptos asociados al negocio que ayudarán a su comprensión y algunas de las soluciones existentes en el mercado internacional que pueden servir de referencia. Finalmente, se describen y caracterizan las tecnologías y herramientas que se utilizarán en el desarrollo de la solución propuesta.

Capítulo 2: Se definen y describen los artefactos generados por los diferentes flujos de trabajo de la metodología de desarrollo utilizada. Además, se describe el algoritmo de funcionamiento que tendrá la solución.

Capítulo 3: Contiene artefactos propios de la implementación de la solución y las pruebas realizadas para la validación del producto final junto a un muestreo de los resultados obtenidos en las distintas iteraciones.

CAPÍTULO 1: Fundamentación Teórica

1.1 Introducción

En este capítulo se brinda información del estado actual de las bibliotecas y aplicaciones que realizan identificación de televisoras, así como las diferentes técnicas que pueden ser utilizadas para esto. Se abordan temas referentes a las imágenes y videos digitales.

1.2 Conceptos asociados al dominio del problema

1.2.1 Imagen digital

Una imagen digital es una matriz de $M \times N$ elementos que se obtiene mediante la discretización¹ de la imagen analógica, tanto en las dimensiones geométricas, mediante la generación de celdas por muestreo equiespaciado de la superficie, como en sentido radiométrico, mediante la cuantificación **(Gonzalez, y otros, 2002)**.

Imagen digital es la representación bidimensional de una imagen empleando bits. De acuerdo a la resolución que presenta la imagen, estática o dinámica, se podrá hablar de gráfico rasterizado² (o mapa de bits; representa una rejilla rectangular de píxeles o punto de color, la cual se puede visualizar en un monitor, en un papel o en cualquier otro dispositivo de representación que se emplee) o de gráfico vectorial (imagen producto de objetos geométricos independientes) **(Definición ABC, 2007-2013)**.

Se puede concluir que una imagen digital no es más que el resultado de codificar y cuantificar un muestreo de señal analógica. La matriz bidimensional obtenida a partir de este proceso estará conformada por bits y sus dimensiones serán definidas por medio de píxeles.

1.2.2 Video digital

El video digital es un tipo de sistema de grabación de video que funciona usando una representación digital de la señal de video. Es el resultado de la combinación de secuencias de imágenes, audio, texto y movimiento que se obtienen por muestreo y cuantificación de la señal de video analógica **(Bertini, y otros, 2005)**.

¹Extraer de un conjunto infinito de puntos una cantidad finita, de manera que aumentando el número de puntos, el conjunto finito tienda al continuo.

²Proceso por el cual una imagen descrita en un formato gráfico vectorial se convierte en un conjunto de píxeles o puntos para ser desplegados en un medio de salida digital.

El video digital consiste en traducir la intensidad y el color en números. Básicamente representa la cantidad de rojo, verde y azul (los tres colores necesarios para reconstruir todos los demás) mediante tres números o dígitos. Digitalizar la señal audiovisual consiste en convertirla en series de números, unos y ceros, de modo que la intensidad luminosa de cada punto queda descrita por dichos números **(Bartolome, 2008)**. Al analizar el área en que se desarrolla la presente investigación y por tener puntos en común con ella se asumirá en lo adelante como concepto de Video Digital.

1.2.3 Marcas de agua

La marca de agua digital es un código de identificación que se inserta directamente en el contenido de un archivo multimedia (imagen, audio, video), de manera que sea difícil de apreciar por el sistema perceptual humano, pero fácil de detectar usando un algoritmo dado y una clave, en un ordenador **(Orúe López, 2007-2012)**.

La marca de agua es un código digital embebido en video y se puede utilizar para la transmisión de información binaria la cual indica por lo general al propietario del material audiovisual **(Hartung, y otros, 2002)**. Es un sistema de protección muy antiguo utilizado por la esteganografía, el cual consiste en insertar un mensaje o un sello de identificación (visible o no) dentro de un objeto digital, ya sea imágenes, documentos de texto, vídeos, etc., con la finalidad de proteger su origen y autenticidad **(Instituto Nacional de Tecnologías de la comunicación, 2013)**.

Las marcas de agua están destinadas a brindar información sobre la identidad del autor o propietario de determinado producto audiovisual. Deben poseer importantes características como imperceptibilidad, la robustez y ausencia de ambigüedad.

1.2.4 Patrón

Es considerado un arreglo de descriptores, que a su vez son los elementos que caracterizan las formas que definen una imagen digital **(Gonzalez, y otros, 2002)**.

A partir de la segmentación de una imagen en diferentes objetos o regiones se define patrón como cada uno de estos objetos que poseen características propias representadas por medio de valores medibles **(Acharya, y otros, 2005)**.

Los patrones son aquellos elementos que conforman y definen una imagen a partir de los descriptores que la conforman. Son elementos valiosos a la hora de realizar búsquedas e identificaciones de objetos en un video.

1.2.5 Logo

Un logotipo es básicamente una configuración realizada a base de letras o dibujos, o ambos, usada para representar a un organismo, empresa o actividad **(Rodríguez Valcarcel, 2006)**.

En lo adelante se trabajará el término logo teniendo como referencia el último de los conceptos definidos por su alcance y claridad.

1.2.6 Televisora

Se llama televisora a aquella estación que emite señales de audio y de video a los aparatos receptores de televisión dentro de un área geográfica determinada **(DefiniciónABC, 2007-20013)**.

1.3 Objeto de Estudio.

1.3.1 Descripción General

Existen diversas técnicas para la identificación automática de elementos presentes en transmisiones audiovisuales. Estas técnicas con diferentes métodos hacen uso de los descriptores presentes en la imagen para revelar información útil en la generación de metadatos. Las prácticas definidas dentro del campo de detección de patrones son muy funcionales a la hora de recopilar datos ya sea una señal analógica o digital.

La búsqueda de patrones es una técnica de análisis que se puede aplicar en detección de objetos, reconocimiento, seguimiento y correspondencia. Puede darse el caso que sea necesario encontrar un patrón dentro de un repositorio de imágenes mucho más complejas en relación con el contenido que representa, sin necesidad de hallar una correspondencia exacta en caso de requerirse, teniéndose en cuenta para ello un grado de variación en el patrón **(García Mateos, 2006)**.

1.3.2 Marcas de agua

Es un código digital contenido en el video que se puede utilizar para la transmisión de información binaria y que indica típicamente el autor del producto audiovisual. Algunos de los requerimientos más importantes para su uso son:

- Invisibilidad: La marca de agua digital debe ser invisible para el ojo humano.
- Seguridad: Debe ser imposible retirar la marca de agua sin autorización.
- Robustez: No se podrá manipular la marca de agua si al mismo tiempo afecta de alguna forma la calidad de la imagen reduciendo su valor comercial.
- Complejidad: En el caso de que la operación de recuperación sea compleja, con el fin de dar cuenta de todas las clases posibles de ataques el marcado debe ser de baja complejidad.
- Velocidad binaria constante (bit-rate): La marca de agua en el dominio de flujo de bits no debería incrementar la tasa de bits, por lo menos para la velocidad binaria constante de aplicaciones en las que el ancho de banda del canal de transmisión ha de ser estándar (Hartung, y otros, 2002).

Aunque existe un movimiento creciente en la industria y la comunidad científica para desarrollar una tecnología de marcas de agua aún hay elementos por resolver, bases teóricas de las investigaciones incompletas y la heurística como factor común en los sistemas diseñados. También la ausencia de normas para evaluar los sistemas de marcas de agua ha evitado que se estandarice un sistema único que permita su uso de forma coherente.

1.3.3 Integrales proyectivas

Se pueden usar en detección, seguimiento y segmentación. Normalmente como fase previa a otros procesos. La principal característica es la reducción de dimensiones de imágenes 2D a proyecciones 1D. Son más rápidas de procesar pero se puede perder información relevante. La técnica es aplicable cuando los objetos de interés son distinguibles por intensidad. Si esto no es posible, también se pueden aplicar las proyecciones sobre imágenes de bordes.

Ventajas:

- Se trabaja con información simplificada y acumulada. Más eficiencia y menos sensibilidad a ruido.

Problemas:

- En algunas aplicaciones, la proyección puede suponer perder información relevante.
- Difícil establecer umbrales máximos y mínimos adecuados (**García Mateos, 2006**).

Aunque es una técnica útil en seguimiento y detección las dificultades que presenta podrían potenciar una serie de falsos positivos o negativos a la hora de buscar los posibles elementos identificadores de la televisora.

1.3.4 Template Matching

Template Matching se utiliza para muchas aplicaciones en el procesamiento de imágenes. La correlación cruzada es el enfoque estadístico básico para el registro de imágenes. Se utiliza para la coincidencia de plantilla o de reconocimiento de patrones. Una plantilla puede ser considerada una sub-imagen de la imagen de referencia, y ésta sub-imagen ser considerada como una imagen detectada. El objetivo es establecer la correspondencia entre la imagen de referencia y la imagen detectada (Image Registration by Template Matching Using Normalized Cross-Correlation , 2009).

Problemas:

- Muy sensible a cambios de escala, rotación o deformaciones 3D de los objetos.
- La aplicación de la técnica es muy costosa. Cuando la resolución aumenta al doble, el tiempo se multiplica por 16 (**García Mateos, 2006**).

A partir de investigaciones realizadas con anterioridad en el Departamento Señales Digitales (**Acosta Jiménez, 2011**) se pueden presentar como posibles soluciones:

- Utilizar varios patrones, con distintos tamaños y rotaciones.
- Hacer una búsqueda multiescala. Aplicar el proceso escalando la imagen a: 50%, 75%, 100%, 125%.
- Usar alguna técnica de atención selectiva. Por ejemplo, usar color o bordes para centrar la atención en ciertas partes de la imagen.

Estas características se pueden utilizar en la detección de elementos claves en la imagen que permitan identificar la televisora que emite la señal. Aunque presenta ciertos problemas a la hora de su ejecución estos pueden ser corregidos con acciones específicas.

La técnica Template Matching se ajusta a las necesidades de la presente investigación. Se considera una buena práctica su uso en el desarrollo de cualquier algoritmo de identificación automática de televisoras puesto que brinda altas posibilidades de efectividad, hay escasa pérdida de información relevante y con la aplicación de las mejoras propuestas su desempeño aumenta notablemente.

1.3.5 Descripción actual del dominio del problema

La identificación de televisoras en el Departamento Señales Digitales es una tarea que incluye diferentes actividades que determinan el flujo del proceso:

Selección del material audiovisual que se analiza o monitorización de una señal televisiva en tiempo real.

- Se trabaja con un material audiovisual previamente grabado o una señal streaming³. En caso de ser streaming la duración de la misma es importante pues definirá el tiempo de monitorización por parte del documentalista. (Si es en vivo, este tiempo no se puede definir)

Identificación de los canales contenidos en la media a partir de elementos distintivos de su programación habitual.

- De forma visual apoyándose en elementos identificables del canal incluidos en su programación (logo, espacios promocionales de la cadena, etc.).
- A partir de su experiencia como observador mediante el reconocimiento de programas habituales de la cartelera de la televisora.
- Consultando un catálogo de referencias confeccionado con información relevante de diferentes televisoras que pueda aportar elementos reconocibles.

Elaboración de un registro de apariciones.

- Documento que permite dos tipos de entradas válidas: el instante de tiempo en que aparece el canal y su nombre así como el instante en que desaparece una televisora y su nombre.

Finalmente la información recogida en el registro es almacenada donde pueda ser consultada para su posterior uso en diferentes soluciones del Departamento Señales Digitales que así lo requieran.

1.3.6 Situación problemática.

El proceso descrito para la identificación de televisoras en las soluciones del Departamento Señales Digitales cumple con el principal requisito definido para esta tarea: generar un documento con la identidad

³ Hace referencia al hecho de transmitir video o audio remotamente a través de una red.

de los canales transmitidos en un material audiovisual junto con sus horarios de inicio y fin de emisión. Sin embargo, a lo largo de su implementación son evidentes vulnerabilidades que evitan el buen desempeño de la tarea y limitan el alcance de sus resultados.

La identificación al ser realizada por usuarios de forma manual requiere recursos en los que clientes potenciales rechazan invertir: numerosas estaciones de trabajo, contratación de empleados solo para que realicen la monitorización y creación de turnos de trabajo especiales para dar seguimiento a transmisiones en tiempo real emitidas en la madrugada. Un gasto como este es poco atractivo y no evita que en ocasiones los registros estén mal redactados y tengan anotaciones inexactas o incompletas. Además, no permite una integración en tiempo real con el resto de las soluciones que utilizarán el metadato que genera su gestión. Existen herramientas que requieren el momento de inicio o fin de aparición de una televisora para realizar catalogaciones de media o grabaciones por eventos en tiempo real. En estos casos dichas herramientas se ven obligadas a una larga espera por el registro de apariciones.

Los problemas de despilfarro de recursos, malgasto de tiempo y demora son elementos que provocan rechazo en los clientes. La automatización del proceso erradicaría en gran medida estas dificultades y permitiría la optimización de otras soluciones.

1.4 Soluciones existentes

Existen varios productos a nivel internacional que realizan el trabajo de identificación de televisoras de forma automática. La razón más importante para desarrollar uno con características similares está dada porque los costos de implantarlos debido a su carácter privativo son altos. A continuación se brinda una muestra de soluciones que contienen funcionalidades para el procesamiento automático de los materiales con los que trabajan.

1.4.1 Teletrax

Ofrece una combinación de marcas de agua de última generación y tecnología de huellas digitales. Con la herramienta se puede evaluar, responder y gestionar información en relación con su contenido en video basado en datos reales. Otras acciones que permite son:

- Evaluar el alcance y el rendimiento de los activos de vídeo
- Confirmar y comprobar emisión de contenido

- Administrar los derechos y la propiedad intelectual
- Control de inventario de contenidos
- Compruebe si hay infracción de derechos de autor

Se encarga de proveer a potenciales clientes información que determina con exactitud cuándo, dónde y por cuánto tiempo su multimedia está siendo utilizada en el mundo televisivo (CIVOLUTION).

1.4.2 Videoma Broadcast Monitor (VDM)

Incorpora la grabación y monitorización multicanal de la señal de Televisión Digital Terrestres (TDT), satelital, de multidifusión, frecuencias analógicas y CCTV (Circuito Cerrado de Televisión). Además de ello realiza el streaming, clipping⁴, catalogación, búsqueda, visionado y almacenamiento del contenido en tiempo real. Cuando las emisiones están provistas de subtítulo procedente del teletexto o de “closed caption”, es posible extraer información textual, incluyendo la transcripción de lo que se está emitiendo (ISID, 2006-2009).

Estas soluciones existentes cuentan con clientes de alto nivel lo que evidencia la calidad de sus propuestas y la amplia gama de espacios que abarca un producto con estas características (Tabla 1).

Civolution (Teletrax)	ISID (VDM)
20th Century Fox	Antena 3 TV
BBC Worlwide	Centro de Satélites de la UE
Sony Pictures Entertainment	Ministerio de Defensa Portugal

Tabla 1 Clientes internacionales para soluciones existentes

El uso de estos productos para dar respuesta a la problemática del presente trabajo no es válido dado su perfil privativo y su falta de compatibilidad con los productos del departamento. Sin embargo, a partir del análisis de los gestores de media mencionados se han reconocido diversas funcionalidades que no pueden faltar en el desarrollo del software para la identificación de canales televisivos. Debe poseer un sistema de capturas, ser capaz de emitir alertas, tanto como gestión de errores como de notificación de eventos, debe permitir al usuario cancelar el proceso de monitorización en cualquier momento y poseer

⁴ Voz inglesa que se traduce como compendio, resumen, recopilación.

una vasta base de datos que le permita identificar la mayor cantidad de televisoras posible. Estos elementos permitirán al producto final estar a la altura de las soluciones internacionales.

1.5 Tecnologías y herramientas para el desarrollo del plugin para la identificación automática de televisoras

En la actualidad existen muchas aplicaciones de software que permiten el procesamiento digital de imágenes y el uso de esta técnica se ha extendido en diversas áreas de la sociedad. Es común verla aplicada en videocámaras digitales de altas prestaciones, sistemas de video-vigilancia o en sencillos equipos domésticos con reconocimiento de objetos (Ej.: cámaras fotográficas que detectan sonrisas) **(Sony, 2012)**.

Existen diferentes herramientas que permiten al desarrollador elaborar una aplicación de este tipo, algunas bibliotecas y lenguajes brindan mejores prestaciones que otros con este fin. Se debe realizar un estudio sobre las tecnologías actuales que permitan el desarrollo de aplicaciones orientadas al procesamiento digital de imágenes. El fin de esta labor es mostrar que tecnologías se utilizarán y la razón por la que se escoge.

1.5.1 Proceso Unificado de Desarrollo de Software (RUP). Metodología de desarrollo

Rational Unified Process (RUP) o Proceso Unificado de Software es una metodología tradicional o pesada. Tiene como objetivo lograr un producto de máxima calidad que cumpla con las necesidades planteadas por el usuario en tiempo y con un presupuesto acordado con anterioridad. Realiza un modelado visual de software y permite gestionar una potente documentación y control de cambios **(Pressman, 2004)**.

RUP es una metodología adaptable a las necesidades de la empresa. Los roles están bien definidos y en caso de ser un proyecto muy grande participan varios equipos con una comunicación bien fluida. El proceso de desarrollo RUP lo divide en cuatro fases, dentro de las cuales se van realizando varias iteraciones, según el proyecto y la importancia de cada actividad que se realiza **(Pressman, 2004)**.

RUP posee tres características esenciales que lo distinguen de las restantes metodologías y lo hacen único:

- Dirigido por casos de uso: Los casos de uso son el claro reflejo de lo que el cliente planteó que deseaba en la modelación del negocio a través de los requerimientos. A partir de que aparecen los

casos de uso, estos pasan a guiar el proceso de desarrollo, ya que los modelos que se obtienen como resultado de la realización de los flujos de trabajo están presentados en los casos de uso.

- Iterativo e incremental: Todas las fases se desarrollan mediante iteraciones. Una iteración incluye actividades de todos los flujos de trabajo, las cuales va refinando en cada iteración.
- Centrado en la arquitectura: La arquitectura muestra una visión común del sistema completo. RUP se desarrolla mediante iteraciones, dando prioridad a los casos de uso arquitectónicamente significativos (**Pressman, 2004**).

Las motivaciones para su uso en el presente trabajo son:

- No depende de la presencia constante del cliente.
- Define una documentación detallada que garantiza la continuidad del proyecto.
- Adaptabilidad a cambios a lo largo del desarrollo de la solución propuesta.

Dentro de los artefactos generados por RUP se utilizarán los siguientes:

- Modelo del Dominio
- Levantamiento de Requisitos
- Diagrama de Casos de Uso
- Descripción de Casos de Uso
- Diagrama de Clases del Diseño
- Diagramas de Secuencia
- Diagrama de Componentes
- Diagrama de Despliegue
- Diseños de Casos de Prueba

1.5.2 Lenguaje Unificado de Modelado (UML).

Es un lenguaje de modelado para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema intensivo.

UML es un lenguaje para modelado de propósito general evolutivo, ampliamente aplicable, soportado por diferentes herramientas e industrialmente estandarizadas. Se aplica a una multitud de diferentes tipos de sistemas, dominios, y métodos o procesos.

- Como lenguaje de *propósito general*, se enfoca en el corazón de un conjunto de conceptos para la adquisición, compartición y utilización de conocimientos emparejados con mecanismos de extensión.
- Como un lenguaje para modelado *ampliamente aplicable*, puede ser aplicado a diferentes tipos de sistemas (software y no - software), dominios (negocios versus software) y métodos o procesos.
- Como un lenguaje para modelado *soportable por herramientas*, las herramientas ya están disponibles para soportar la aplicación del lenguaje para especificar, visualizar, construir y documentar sistemas.
- Como un lenguaje para modelado *industrialmente estandarizado*, no es un lenguaje cerrado, propiedad de alguien, sino más bien, un lenguaje abierto y totalmente extensible reconocido por la industria.

UML posibilita la captura, comunicación y nivelación de conocimiento estratégico, táctico y operacional para facilitar el incremento de valor, aumentando la calidad, reduciendo costos y reduciendo el tiempo de presentación al mercado; manejando riesgos y siendo proactivo para el posible aumento de complejidad o cambio (**Alhir, 1998**).

Las motivaciones para su uso en el presente trabajo son:

- Es ideal para la modelación de los artefactos generados por RUP.
- Permite generar artefactos de acuerdo al sistema de modelado utilizado en el SGPM.

1.5.3 Herramienta de modelado de software

Durante los diferentes ciclos de vida del software la representación del mismo es llevada a cabo por medio de herramientas CASE (Computer Aided Software Engineering). Estas se encargan de realizar diagramas para dicha representación y facilitar las tareas de coordinación para eventos que necesitan ser mejorados en el ciclo de desarrollo.

1.5.4 Visual Paradigm 8.0

Es una herramienta multiplataforma que asimila por completo el ciclo de desarrollo del software. Permite modelar base de datos, requerimientos, proceso de negocio. Pone a disposición del usuario numerosas variantes para realizar diagramas y generar código y documentación a partir de ellos (**Sierra, 2007**).

Características que presenta Visual Paradigm de interés para el presente trabajo:

- Se pueden realizar distintos tipos de diagramas, por ejemplo: Diagramas de clases, de Casos de Uso, de Secuencia y de Componentes. De esta manera responde a las necesidades de cualquier sistema que utilice RUP como metodología de desarrollo.
- Permite realizar ingeniería inversa en el lenguaje C++.
- Exportación en imágenes de los diagramas que son elaborados, con formato jpg, png y svg.
- Genera el código de aplicación según el modelado de la aplicación.
- Genera la documentación de lo que se está modelando.
- Modela todos los diagramas de UML.

1.5.5 Lenguaje de programación C++

El alto consumo de recursos que genera el uso de técnicas y algoritmos de procesamiento digital es un elemento a tener en cuenta a la hora de elegir el lenguaje de programación para desarrollarlas. En el caso del plugin para la identificación automática de televisoras se seleccionó C++ pues aprovecha las potencialidades de la estación donde se programa, posee ventajas en el trabajo de procesamiento y ha sido el escogido por los arquitectos del Departamento Señales Digitales para el desarrollo de aplicaciones de escritorio.

El C++ es a la vez un lenguaje procedural (orientado a algoritmos) y orientado a objetos. Como lenguaje procedural se asemeja al C y es compatible con él, aunque ya se ha dicho que presenta ciertas ventajas. Como lenguaje orientado a objetos se basa en una filosofía completamente diferente, que exige del programador un completo cambio de mentalidad. Las características propias de la Programación Orientada a Objetos de C++ son modificaciones mayores que sí cambian radicalmente su naturaleza.

Los programas escritos en C o C++ tienen otras ventajas sobre el resto. Con la excepción del ensamblador, generan los programas más compactos y rápidos. El código es portable, es decir, un

programa ANSI en C o C++ podrá ejecutarse en cualquier máquina y bajo cualquier sistema operativo. Y si es necesario, proporcionan un acceso a bajo nivel de hardware sólo igualado por el ensamblador **(González, 2008)**.

Las principales características para seleccionar C++ como base del desarrollo de la solución propuesta son:

- Difusión: Posee un gran número de usuarios y existe una gran cantidad de libros, cursos y páginas web dedicados a su estudio.
- Portabilidad: Está estandarizado y un mismo código fuente se puede compilar en diversas plataformas.
- Herramientas: existe una gran cantidad de compiladores, depuradores y librerías para el trabajo con él.
- El sistema al ser terminado se acoplará al SGPM el cual está desarrollado con este lenguaje de programación.
- Facilita el procesamiento de video pues permite optimizar la utilización de recursos de hardware de las computadoras.

1.5.6 Marco de trabajo

En un framework se incluyen soporte de programas, bibliotecas y lenguajes interpretados ayudando al desarrollo y posterior unión de los componentes de una solución.

1.5.7 Qt

Es una biblioteca multiplataforma para desarrollar aplicaciones. Utiliza el lenguaje C++ de forma nativa y permite el desarrollo de aplicaciones con o sin interfaz gráfica. Se distribuye bajo licencia Lesser General Public License (LGPL) y su código es abierto **(Maldonado, 2011)**.

Algunas características de interés para el presente trabajo son:

- Cuenta con métodos para acceder a bases de datos mediante SQL.
- Permite la programación concurrente (Hilos) independientemente del sistema operativo.

- Cuenta con herramientas y clases que permiten acceder a bibliotecas de procesamiento de video.

1.5.8 Entorno de Desarrollo Integrado (IDE). Qt Creator 4.8

Para la implementación del sistema se decidió usar como IDE el Qt Creator debido a las múltiples ventajas para la programación con C++, las cuales se exponen a continuación:

- Tiene plugins para varios sistemas de control de versiones.
- Posee una herramienta de búsqueda eficaz donde se puede buscar fácilmente las clases, métodos y archivos.
- Depurador visual para C++, es consciente de la estructura de muchas clases de Qt, lo que aumenta la capacidad de mostrar los datos de Qt con claridad.
- Se integra perfectamente con el lenguaje C++.

Otras características son:

- Posee un avanzado editor de código C++.
- Herramienta para proyectos y administración.
- Ayuda integrada sensible al contexto.
- Depurador visual.
- Resaltado y auto-completado de código.

1.5.9 Biblioteca para el procesamiento de imágenes. OpenCV 2.1

El desarrollo de un plugin utilizando bibliotecas de clase es atractivo por el ahorro de tiempo que significa el uso de sus funcionalidades previamente implementadas. De esta forma el programador no tiene que realizar las funciones más básicas y garantiza que el producto sea terminado dentro del ciclo definido. OpenCV es una biblioteca open source para C/C++ para procesamiento de imágenes y visión computarizada. Posee estructuras básicas de datos para operaciones con matrices y procesamiento de imágenes como también permite una simple visualización de datos y extracción de información de videos e imágenes. Incluye funcionalidades como: realizar operaciones básicas, procesado de imágenes, análisis estructural, análisis de movimiento, reconstrucción 3d, calibración de la cámara, segmentación y reconocimiento de objetos, entre otras. Sus algoritmos están basados en estructuras de datos muy

flexibles, acoplados con estructuras IPL; más de la mitad de las funciones han sido optimizadas aprovechándose de la Arquitectura de Intel **(Hernández Franco, y otros, 2008)**.

Para el presente trabajo algunas funcionalidades que se utilizan son:

- Presencia de Template Matching como función nativa (cvMatchTemplate).
- Posee funciones para escalado de imágenes, resaltado de bordes y gestión del color (cvDilate, cvCanny, cvCvtColor).
- Permite realizar captura de señales audiovisuales para su posterior manipulación (cvVideoCapture).
- Está orientado para el trabajo con C++, lenguaje seleccionado para el desarrollo del sistema.

Las librerías de OpenCV con las que trabaja el nuevo sistema son:

- cv
- cxcore
- highgui

1.5.10 Sistema Gestor de Bases de Datos (SGBD). Postgre SQL 9.0

La generación y constante demanda de información ha potenciado el uso de las bases de datos. Los Sistemas Gestores de Bases de Datos se encargan de definirlos y permitir la elección de las estructuras de datos necesarias para su gestión.

En la presente investigación se presenta la necesidad de crear una base de conocimientos de logos centralizada que pueda ser utilizada desde varios nodos de tipo PCI. Para crear esta base de datos se utilizará una herramienta extensible y con uso generalizado en varios sistemas del Departamento Señales Digitales, Postgre SQL. Además, Qt contiene una librería para comunicarse de forma simple con este gestor.

Postgre SQL es una herramienta confiable que permite mantener la integridad de los datos. Es un sistema multiplataforma que posee características como las subconsultas, las restricciones a valores en los campos y los valores por defecto **(Douglas, y otros, 2003)**.

Posee una integridad referencial e interfaces nativas para varios lenguajes, entre ellos C++. Postgre SQL se puede modificar y distribuir de forma gratuita para cualquier fin gracias a su licencia liberada.

1.6 Conclusiones parciales

El estudio del marco teórico en el que se presenta esta investigación, así como de los diferentes productos del mercado que incorporan la identificación de televisoras arrojó los siguientes elementos de interés:

- Una solución foránea a los productos del Departamento Señales Digitales no es necesaria. La investigación realizada hasta el momento demuestra que esto encarecería el negocio y limitaría su comercialización por sus características privativas.
- Las tecnologías y herramientas estudiadas a lo largo del capítulo permiten el desarrollo de un plugin capaz de procesar media e integrarse a la PCI bajo los mismos estándares utilizados en el SGPM.
- De las diferentes técnicas automáticas para la identificación de televisoras es el Template Matching con su búsqueda de patrones el más adecuado para los intereses de la presente investigación.
- Se deben tener en cuenta las limitaciones de la técnica Template Matching, como son los cambios de escala para poder actuar en consecuencia y solucionarlas.

CAPÍTULO 2: Análisis y diseño

2.1 Introducción

En este capítulo se realizará una descripción del diseño del plugin para la identificación automática de televisoras propuesto como solución a la problemática planteada en el Capítulo 1. Se definirá el modelo del dominio, la descripción de las clases que lo componen, requisitos funcionales y no funcionales correspondientes al componente en desarrollo, así como casos de usos y actores definidos. Para la mejor comprensión de cada funcionalidad solo se describen los casos de usos críticos y de esta forma llegar a partir de su análisis a una solución que cumpla con los requisitos funcionales definidos.

2.2 Modelo del Dominio

La ausencia de un negocio real y por consiguiente la imposibilidad de precisar la estructura de los procesos de negocio son las razones por las que se realiza un modelo del dominio. Este modelo (**Fig. 1**) es el comienzo necesario para alcanzar el diseño adecuado del sistema a desarrollar pues expone la lógica obtenida del problema. Representa las clases conceptuales del problema real y no de componentes virtuales. Facilita un panorama visual de los principales conceptos que se utilizan en el dominio del problema que junto con un glosario de términos permiten una mejor comprensión de la solución.

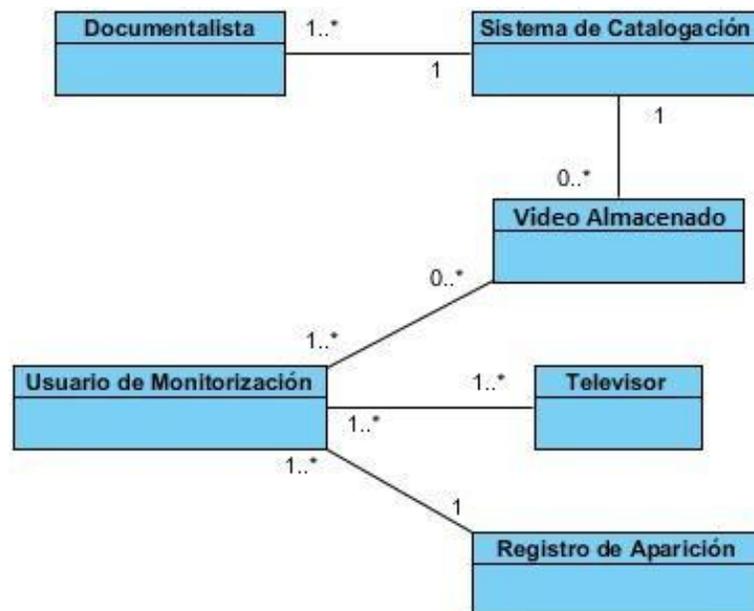


Figura 1 Modelo del Dominio

2.3 Descripción del Modelo del Dominio

El modelo del dominio describe el proceso de identificación de televisoras en dos momentos: cuando este es realizado a través del Sistema de Catalogación o SC (producto perteneciente al proyecto Sistema de Captura y Catalogación de Media) y cuando es llevado a cabo a través de una monitorización televisiva.

- En el primer caso un documentalista accede a través del SC a un video almacenado con anterioridad para registrar todas las posibles propiedades del mismo, entre ellas el nombre del canal televisivo que lo transmitió originalmente, considerándose esta acción como un tipo de identificación.
- En el segundo caso un usuario de la monitorización puede acceder a un video directamente o controlar una transmisión televisiva en tiempo real. En ambos casos mientras se lleva a cabo la visualización del material se documenta en el Registro de Aparición los ocultamientos y apariciones de las televisoras.

2.4 Descripción de las clases del Modelo del Dominio

Documentalista: A través del sistema de catalogación inicializa un video para obtener sus metadatos. Dentro de los metadatos que genera está el nombre de la televisora que transmitió la señal.

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

Video almacenado: Son capturas de video almacenadas para ser consultadas por documentalistas o usuarios de monitorización.

Usuario de monitorización: Se encarga de dar seguimiento al contenido de una media o de una señal televisiva para identificar los momentos de aparición y ocultamiento de las televisoras. Posteriormente se encarga de incluir esta información en un registro de apariciones.

Televisor: Es el medio a través del cual el usuario de monitorización es capaz de visualizar una señal de video transmitida en tiempo real.

Registro de aparición: Es el documento que contiene los momentos de aparición y ocultamiento de las televisoras.

2.5 Requisitos

Los requisitos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas (**Sommerville, 2005**). O sea, las condiciones necesarias que debe cumplir el sistema para alcanzar los objetivos propuestos.

2.5.1 Requisitos Funcionales

Los requisitos funcionales de un sistema describen lo que este debe hacer y los recursos específicos que debe proporcionar. En el caso del Plugin para la Identificación Automática de Televisoras se han definido los siguientes:

R1. Monitorizar televisora

Descripción: El plugin debe ser capaz de monitorizar una señal definida por la plataforma a través de una dirección URL. Simultáneo a este proceso se realizará una búsqueda de patrones en la imagen con el fin de detectar un logo y una vez descubierto compararlo con una base de datos existente e identificar la televisora que representa. Este proceso se repetirá cada vez que se detecten posibles cambios en la identidad de la televisora.

Entradas:

- Dirección origen del archivo de video (cadena de string) o Dirección origen de la señal streaming (cadena de string).

Salidas:

- Notificación de televisora identificada (instante de tiempo en que aparece [hh:mm:ss] – nombre de la televisora [cadena de string]).
- Notificación de desaparición de televisora (instante de tiempo en que desaparece [hh:mm:ss] – nombre de la televisora [cadena de string]).

R2. Anotar aparición de televisora

Descripción: Al ser identificada una televisora la plataforma debe recibir información de este evento.

Entradas:

- Notificación de televisora identificada

Salidas:

- Instante de tiempo en que se identificó
- Nombre de la televisora

R3. Anotar desaparición de televisora

Descripción: Al ser sustituida una televisora por una nueva la plataforma debe recibir información de este evento.

Entradas:

- Notificación de cambio de identidad.

Salidas:

- Instante de tiempo de sustitución
- Nombre de la televisora sustituida

R4. Brindar porciento de proceso.

Descripción: La plataforma debe recibir el estado de la monitorización a través del porciento ejecutado de la misma por parte del plugin.

R5. Notificar error

Descripción: La plataforma recibirá un reporte cada vez que ocurra un error en el desempeño del plugin.

R6. Detener proceso.

Descripción: Permite al sistema detener el proceso de monitorización en cualquier momento.

2.5.2 Requisitos No Funcionales

Describen cómo el software cumplirá con sus funciones. Al ser difíciles de verificar son evaluados subjetivamente. Constituyen la forma de trabajo del sistema para que su desempeño sea eficiente (**Sommerville, 2005**). Elementos como la disponibilidad y el diseño son comunes para su definición.

R1. Soporte

Descripción: La base de datos que contiene los logotipos para la identificación de las televisoras debe permitir agregar en cualquier momento nuevos elementos.

R2. Disponibilidad

Descripción: La base de datos que contiene los logotipos para la identificación de las televisoras debe estar disponible desde cualquier estación de trabajo.

R3. Restricciones en el diseño e implementación

Descripción: La construcción del sistema debe corresponderse con la utilizada en el SGPM para lograr una exitosa integración con la PCI. Es necesario el uso del lenguaje C++ junto con Qt.

2.6 Algoritmo de funcionamiento para el plugin identificador de televisoras.

Para la elaboración del plugin una de las técnicas que se utilizará es el Template Matching, esta técnica presenta problemas en su desempeño cuando existen cambios de escala. Para dar solución a esa problemática se decide el uso de una base de datos que contenga metadatos asociados a la imagen de un logo. Uno de los metadatos almacenados en esta base de conocimientos será la resolución del video del que fue extraída la plantilla, de esta forma el algoritmo en su ejecución podrá utilizar esta información para escalarla con bastante aproximación a la resolución del video que se esté monitorizando.

Con las precisiones mencionadas el algoritmo se define de la siguiente manera:

1. Extraer fotograma del video mediante las funciones que brinda el OpenCV.
2. Establecer las regiones de interés del fotograma para limitar la búsqueda en esta área.
3. Aplicar filtros a la imagen con el objetivo de mejorar la visibilidad del logo en la misma (Bordes, Dilatación, Escalado en caso de ser necesario).
4. Aplicar Template Matching al fotograma utilizando cada uno de los logos en BD hasta encontrar una coincidencia.
5. Si se encontró notificar como aparición de televisora. Si no volver al paso 1
6. Capturar próximo fotograma hasta que no se detecte el logo.
7. Cuando no se detecte el logo notificar tiempo de ocultamiento de la televisora.
8. Volver al paso 1 hasta que finalice el video.

Este algoritmo realiza un análisis exhaustivo, compara para cada fotograma los elementos detectados en las regiones de interés con los logos almacenados en la base de datos. De esta forma aunque en el video estén contenidas señales de varias televisoras el plugin podrá identificarlas y notificar sus instantes de aparición y ocultamiento.

2.7 Diagramas de Caso de Uso del Sistema (DCUS)

Un caso de uso es una descripción de las tareas que deben ser realizadas para que un proceso se ejecute. Se reconoce también como la secuencia de interacciones que existirán entre un sistema y sus actores respondiendo a un evento inicializado por un actor principal sobre el sistema (Jacobson, et al., 2004). El diagrama que representa el caso de uso del sistema para la presente investigación es el siguiente (**Fig. 2**):

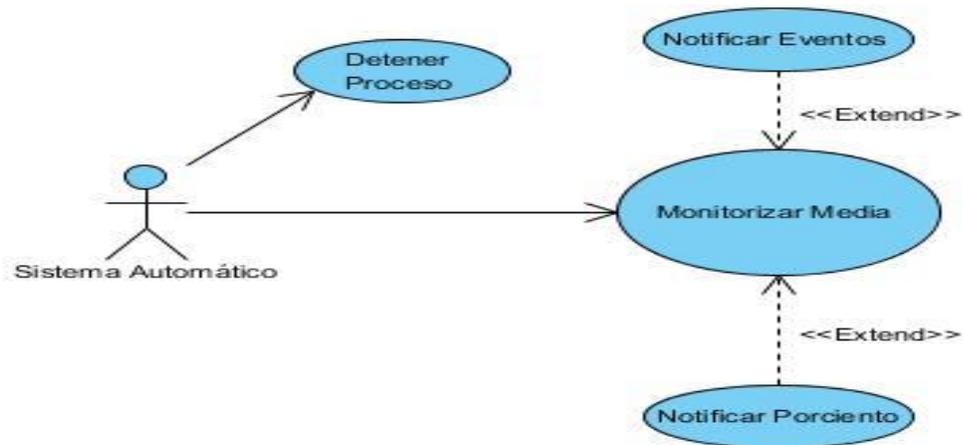


Figura 2 Diagrama de Casos de Uso del Sistema

2.7.1 Definición de los actores

Actor	Descripción
Sistema Automático	Inicializa la ejecución del proceso de monitorización con el envío de una dirección URL o señal streaming con la ubicación de la media que se utilizará. Es capaz de detener el proceso en cualquier instante.

2.7.2 Especificación de los Casos de Uso del Sistema (CUS)

Objetivo	Identificar televisoras presentes en la señal que se monitorizará.
Actores	Sistema Automático: (Inicia) y envía la dirección donde estará ubicada la media a monitorear.
Resumen	Es inicializado al recibir las direcciones de la media a monitorizar. Se validan las entradas que recibe del sistema y realiza una búsqueda de patrones sobre la señal para detectar la presencia de un logo y utilizarlo en la identificación de la televisora.
Complejidad	Alta
Prioridad	Crítico
Precondiciones	Debe recibir una dirección con la ubicación de un archivo de video.
Postcondiciones	Se inicializan los CU Notificar Eventos y Notificar Porciento si se cumplen determinadas condiciones: la existencia de identificaciones positivas y que la media que se analiza tenga definido un tiempo de

	duración respectivamente.	
Flujo de eventos		
Flujo básico: Monitorizar		
	Actor	Sistema
1.	Envía el identificador del proceso que se ejecuta y la dirección URL con la ubicación del archivo de video que se analizará.	
2.		Comprueba la veracidad de los datos enviados por el sistema automático.
3.		Aplica un algoritmo de identificación de patrones a la media para detectar el logo del canal que se visualiza.
4.		Si encuentra un logo lo captura para usarlo como elemento comparativo.
5.		Realiza una comparación entre la captura y logos asociados a televisoras almacenados en una base de datos perteneciente al sistema.
6.		En caso de existir un resultado positivo se envía al CU Notificar Evento el instante de tiempo en que aparece el logo y el nombre de la televisora asociada al mismo.
7.		En caso de desaparecer un logo ya notificado y surgir uno nuevo se envía al CU Notificar Evento el instante de tiempo en que desaparece el logo y el nombre de la televisora asociada al mismo.
8.		En caso de que la media seleccionada posea un tiempo de

		duración esta información será enviada al CU Notificar Por ciento. Inicializa CU Notificar Por ciento.
Flujos alternos		
2a. La dirección recibida no es válida.		
	Actor	Sistema
2a.		Notificar al Sistema Automático el mensaje: "La dirección origen no existe o no se tiene permisos de lectura sobre ella. Termina el CU.
Flujos alternos		
5a. El sistema no se conecta a la BD.		
	Actor	Sistema
5a.		Notificar al Sistema Automático el mensaje: "No se puede acceder a la Base de Datos". Termina el CU.
Relaciones	CU Incluidos	
	CU Extendidos	Para quien publique: Notificar Evento: Paso 6 del Flujo Básico Notificar Porciento: Paso 8 del Flujo Básico Para quien extienda: Notificar Evento: en el Monitorizar Media Notificar porciento: en el Monitorizar Media
Requisitos no funcionales	RN 1 y RN 2	
Asuntos pendientes		

Tabla 2 Descripción del CUS Monitorizar Media

Objetivo	Notificar al Sistema Automático la ocurrencia de eventos propios del sistema.
Actores	Sistema Automático: (Inicia).
Resumen	Permite notificar al Sistema Automático el nombre de la televisora

	identificada así como el instante de tiempo en que se detecta por primera vez o el instante de tiempo en el que se deja de emitir su señal.	
Complejidad	Baja	
Prioridad	Normal	
Precondiciones	El CU será iniciado cada vez que sea identificada o sustituida una televisora.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.		Recibe los datos: nombre de la televisora identificada así como instante de tiempo en el que se identificó.
2.		Envía una cadena de caracteres al Sistema Automático con la información recibida (nombre de la televisora-hh/mm/ss).
3.		Recibe los datos: nombre de la televisora que desaparece así como instante de tiempo en el que lo hace.
4.		Envía una cadena de caracteres al Sistema Automático con la información recibida (nombre de la televisora-hh/mm/ss).
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales	No existen	
Asuntos pendientes		

Tabla 3 Descripción del CUS Notificar Evento

Objetivo	Brindar un resumen sobre el estado de completamiento del proceso.	
Actores	Sistema Automático: (Inicia).	
Resumen	Contempla la notificación al sistema del avance porcentual del proceso de monitorización.	
Complejidad	Baja	
Prioridad	Normal	
Precondiciones	La media que se analiza debe tener tiempo de duración.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
1.		Recibe el dato: tiempo de duración.
2.		Calcula el porcentaje a partir del tiempo de duración recibido y el instante en el cual se encuentra la monitorización para mostrarlo en una barra de progreso.
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales	No existen	
Asuntos pendientes		

Tabla 4 Descripción del CUS Notificar Porcentaje

Objetivo	Detener el proceso de monitorización
Actores	Sistema Automático: (Inicia).
Resumen	Permite al sistema detener el proceso de monitorización en cualquier

	momento.	
Complejidad	Baja	
Prioridad	Media	
Precondiciones	Tiene que estar ejecutándose el proceso de monitorización.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
3.		Recibe solicitud de detención.
4.		Detiene el proceso de monitorización.
Flujos alternos		
	Actor	Sistema
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales	No existen	
Asuntos pendientes		

Tabla 5 Descripción del CUS Detener Proceso

2.8 Arquitectura del sistema

La arquitectura es la estructura de estructuras de un sistema, la cual abarca componentes de software, propiedades externas visibles de estos componentes y sus relaciones (**Bass, et al., 2003**), es el nivel más alto de un sistema en su ambiente.

El procesamiento de video requiere alta capacidad de cómputo, en este caso es ideal el uso de la arquitectura para un sistema distribuido, ya que permite que el procesamiento de la información se distribuya sobre varios ordenadores en vez de estar confinado solamente en uno (**Sommerville, 2005**).

Para tener una visión clara del funcionamiento de la arquitectura del SGPM, se presenta en la Fig. 3 la estructura en forma de pizarra de control que presenta el diseño. El mismo permite organizar y estructurar cómputos, así como software que se encargan de realizar tareas de procesamiento sobre materiales audiovisuales.

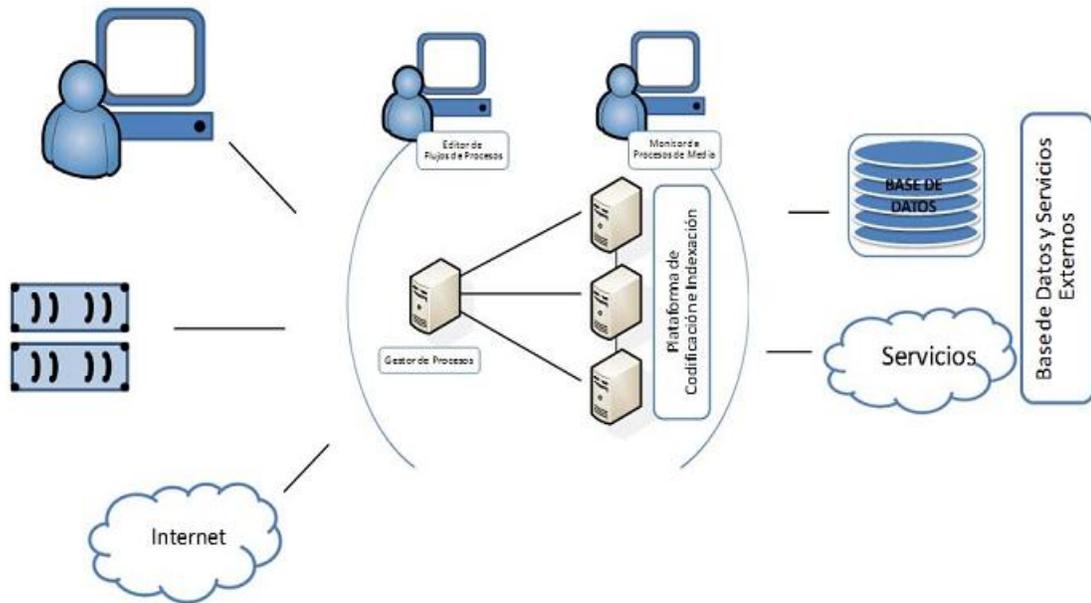


Figura 3 Estructura del Sistema Gestor de Procesos de Media

Para el desarrollo del Sistema Gestor de Procesos de Media es necesario hacer funcionar como un único sistema al Gestor de Procesos de Media (GPM), Monitor de Procesos de Media (MPM) y la granja de servidores dedicados que pueden ser diversos **(Suárez Pérez, y otros, 2012)**.

En el desarrollo del plugin se tiene en cuenta que se integrará al SGPM a través de la PCI por lo que la arquitectura que se utiliza es la definida para estos productos.

Uno de los elementos que la caracterizan son:

- Uso del estilo en capas, estructurando la aplicación en varias capas, lo que permitirá llevar el desarrollo a varios niveles y delimitar bien las responsabilidades y el alcance de cada parte del software.

Las ventajas que ha presentado la aplicación de dicha arquitectura son varias **(Suárez Pérez, y otros, 2012)**:

- Balance de cargas equilibrado haciendo uso de una de las técnicas más usadas en
- ambientes distribuidos.
- La arquitectura plantea alto grado de escalabilidad permitiendo se incrementen de manera sencilla nodos de procesamiento a la granja de esclavos o cómputos dedicados.

- Permite que se utilice la técnica de réplica de servidores y funcionalidades, logrando de esta forma que el sistema sea muy tolerante a los fallos.

En el caso de la PCI como sistema al cual se integrará el plugin su desarrollo se llevó a cabo con la utilización de técnicas de programación multihilos aplicando la arquitectura de dos capas y la implementación de plugins (**Cabrera Martínez, 2011**).

2.9 Patrones de diseño

Soluciones estándar que brindan una serie de ventajas a programadores y diseñadores del sistema. Dan respuesta a un problema común en cuanto a diseño de sistemas e implementación. Facilita la comunicación entre diseñadores, la reutilización de código así como la conformación de un diseño con posibilidades de cambio. Se aplican a un elemento específico, como un agregado de componentes para resolver algún problema de diseño, relaciones entre los componentes o los mecanismos para efectuar las comunicaciones entre componentes. Los patrones intentan codificar el conocimiento, las expresiones y los principios ya existentes. Estos brindan una serie de recomendaciones para organizar los distintos componentes del sistema y así tener un mejor entendimiento del problema al que se le quiere dar solución.

Para conformar el modelo de diseño fueron utilizados los patrones GRASP: Experto, Creador, Bajo acoplamiento, Alta cohesión y Controlador. También fueron utilizados los patrones GoF: Observador y Fachada.

2.9.1 Patrones GRASP (General Responsibility Assignment Software Patterns)

Los patrones de asignación de responsabilidades describen los principios fundamentales de la asignación de responsabilidades a objetos, expresado en forma de patrones (**Larman, 2005**).

- **Experto:** El patrón Experto propone la asignación de la responsabilidad de experto a la clase que cuenta con la información necesaria para cumplir con la misma. Si esta asignación se hace de manera correcta se puede entender el sistema de una mejor manera siendo más fácil de mantener y ampliar, brindando la oportunidad de reutilizar sus componentes en futuras aplicaciones. **Uso en el sistema:** Clase Identificar Televisora al ser la que inicializa el proceso de identificación.
- **Creador:** El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos. Se aplicará en todos los casos donde una clase tiene la responsabilidad de crear una nueva

instancia de la otra. La correcta asignación permite que la aplicación pueda soportar un bajo acoplamiento, una mayor claridad, el encapsulamiento y reutilización. **Uso en el sistema:** En el caso de la clase Identificar Televisora se necesita inicializar un objeto del tipo Hilo Identificar para poder dar comienzo al método Anotación Exhaustiva. De la misma forma hace Hilo Identificar con la clase Logo al ser necesario inicializar los logos presentes en la Base de Datos.

- **Bajo acoplamiento:** El patrón de Bajo acoplamiento estimula a asignar responsabilidades de modo tal que no influya en que las dependencias de una clase sean de muchas otras. Tener un bajo acoplamiento soporta el diseño de clases más independientes que reducen el impacto de los cambios y permite una mayor reutilización de código. **Uso en el sistema:** Este patrón se utilizó en todas las clases del sistema propuesto.
- **Alta cohesión:** Se ocupa de que las clases del diseño realicen las funcionalidades necesarias para cumplir con las tareas que tienen definidas. Plantea la contribución entre clases para realizar tareas de elevada complejidad. **Uso en el sistema:** Este patrón se utilizó en todas las clases del sistema propuesto.
- **Controlador:** Define quién deberá encargarse de atender un evento del sistema. Es una clase que para el diseñador representa de alguna manera al sistema global. **Uso en el sistema:** Este patrón se evidencia a través de la clase Identificar Televisora.

2.9.2 Patrones GoF (Gang of Four)

Son agrupados en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento. Los estructurales tratan la combinación de clases, su relación y la formación de estructura de alta complejidad, mientras que los creacionales tratan la creación de instancias y los de comportamientos tratan la interacción y la cooperación entre clases (Larman, 2005).

- **Observador:** Define una dependencia “uno-a-muchos” entre objetos, para que, cuando uno de ellos cambie su estado, todos los que dependan de él sean avisados y puedan actualizarse convenientemente. Esto se puede notar en la relación existente entre la clase controladora del plugin y el medio observador de la PCI.

- **Fachada:** Provee de una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema. En el caso del plugin se evidencia cuando se crea una relación entre él y la PCI.

2.10 Modelo del Diseño

Al modelar el sistema en el diseño se debe tener en cuenta el soporte a todos los requisitos, de esta forma se logrará una descripción detallada de los requisitos y una estructura del sistema. El modelo del diseño es un modelo de objetos que describe la ejecución real de los casos de uso enfocándose en como los requisitos (funcionales y no funcionales) tienen impacto en el sistema. Funciona también como una entrada para las actividades de implementación (Jacobson, et al., 2004).

2.10.1 Diagrama de Clases del Diseño

Describe de forma gráfica los detalles de las clases e interfaces en la aplicación. Se conforma con elementos como: clases, atributos, interfaces, métodos, tipos de atributos y dependencias.

2.10.2 Diagrama de Clases del Diseño para el Plugin para la Identificación de Televisoras

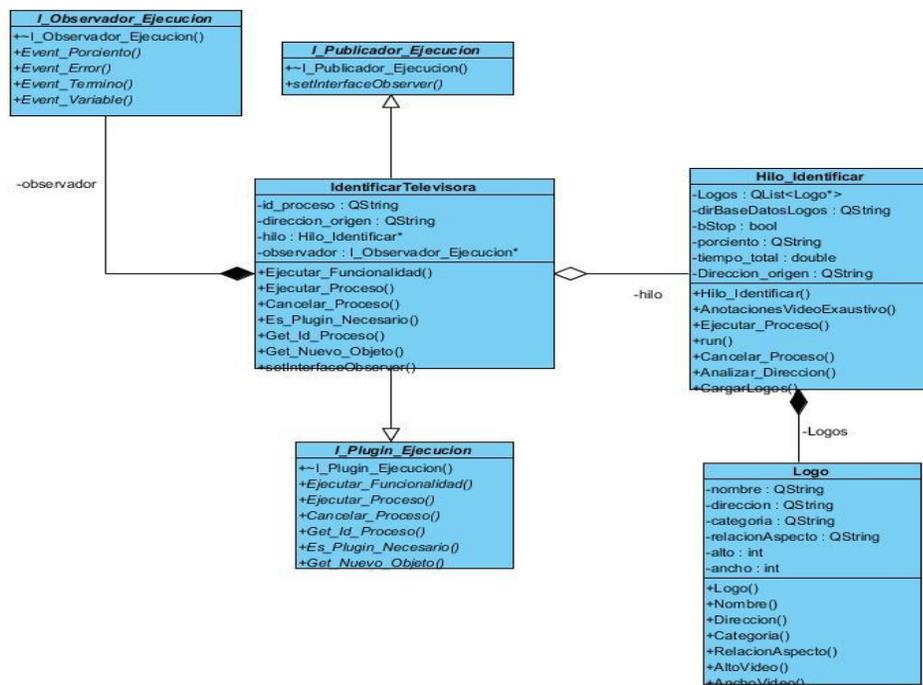


Figura 4 Diagrama de Clases del Diseño

2.11 Diagramas de secuencia

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso, contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

2.11.1 Diagrama de secuencia CU Monitorizar Media

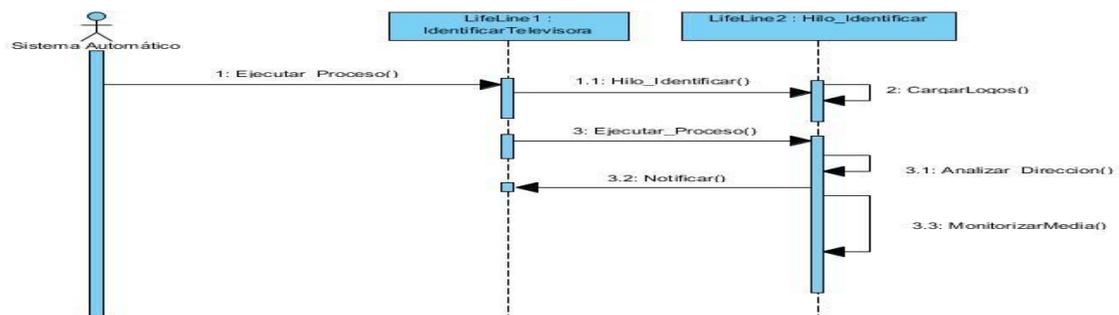


Figura 5 Diagrama de secuencia CU Monitorizar Media

2.11.2 Diagrama de secuencia CU Notificar Evento

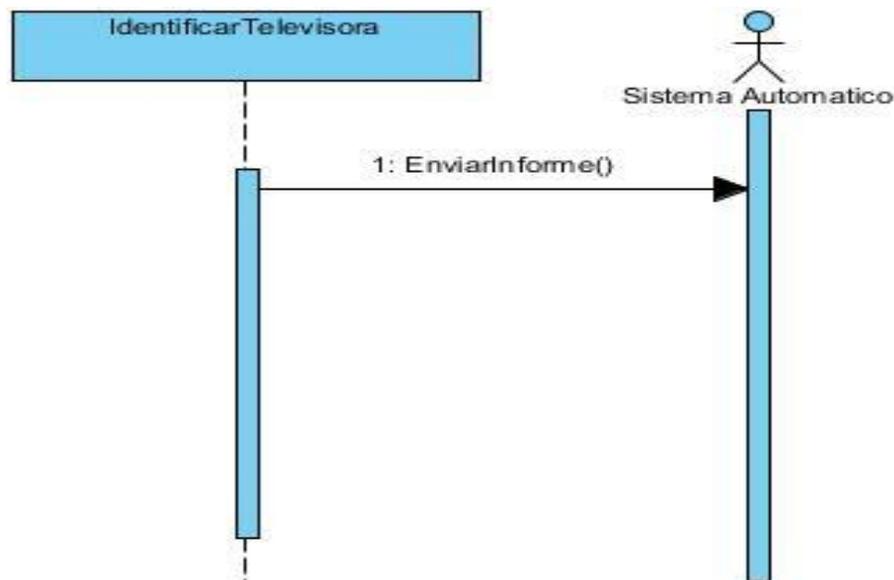


Figura 6 Diagrama de secuencia CU Notificar Evento.

2.11.3 Diagrama de secuencia CU Notificar Porciento

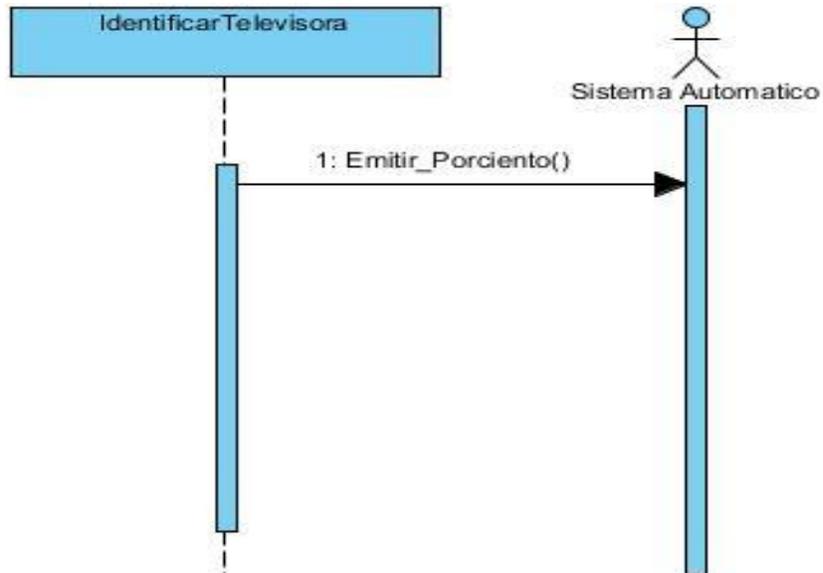


Figura 7 Diagrama de secuencia CU Notificar Porciento

2.11.4 Diagrama de secuencia CU Detener Proceso

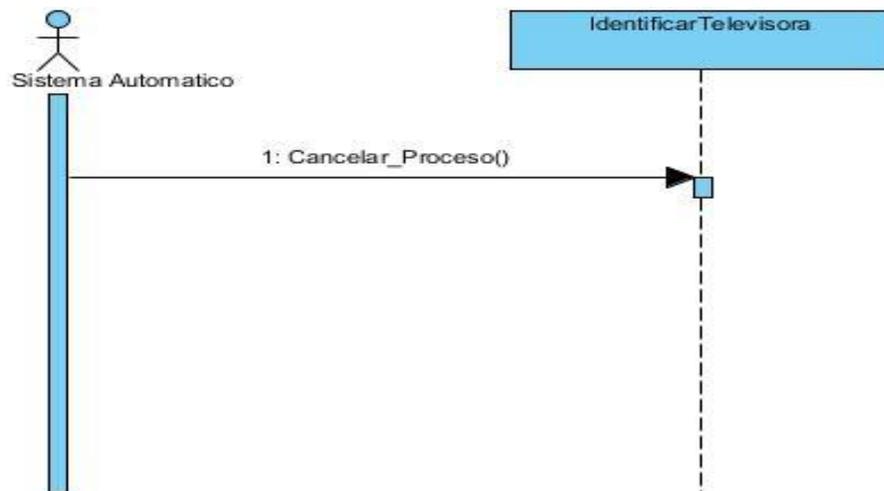


Figura 8 Diagrama de secuencia CU Detener Proceso

2.12 Descripción de las clases del diseño

Nombre	Identificar Televisora	
Tipo	Controladora	
Atributos	Descripción	

id_proceso	QString
direccion_origen	QString
hilo	Hilo_Identificar
observador	I_Observador_Ejecucion
Para cada responsabilidad	
Nombre	Ejecutar_Funcionalidad()
Descripción	Ejecutar una funcionalidad cualquiera sobre un proceso de ejecución. Se hereda de I_Plugin_Ejecucion.
Nombre	Ejecutar_Proceso()
Descripción	Construir el hilo de ejecución que va a realizar el monitoreo.
Nombre	Cancelar_Proceso()
Descripción	Detiene el proceso de monitorización.
Nombre	Es_Plugin_Necesario()
Descripción	Verifica que el plugin se ejecute sobre los parámetros que se están enviando.
Nombre	Get_Id_Proceso()
Descripción	Retorna el identificador del proceso necesario para su posterior reconocimiento por parte de la Plataforma.
Nombre	Get_Nuevo_Objeto()
Descripción	Devuelve una copia del plugin para que sea almacenado en memoria.
Nombre	setInterfaceObserver()
Descripción	Es el método utilizado para enviar la información generada por el plugin a la plataforma.

Tabla 6 Clase Controladora Identificar Televisora

Nombre	Hilo_IDentificar
Tipo	Entidad
Atributos	Descripción
logos	QList<Logo*>
dirBaseDatosLogos	QString
bStop	bool
porciento	QString
tiempo_total	double
direccion_origen	QString
Para cada responsabilidad	
Nombre	Hilo_Identificar()
Descripción	El constructor de la clase se encarga de inicializar el objeto, cargando la dirección de la base de datos donde se encuentran los logos del sistema y realiza el llenado de la lista de logos.

Nombre	AnotacionesVideoExhaustivo
Descripción	Método que se encarga de identificar y retornar las variables: momento de aparición de una televisora, momento de ocultamiento y nombre.
Nombre	Ejecutar_Proceso()
Descripción	Da comienzo al monitoreo.
Nombre	Analizar_Direccion()
Descripción	Se encarga de validar la dirección suministrada al proceso como origen de la media a analizar.
Nombre	CargarLogos()
Descripción	Inicializa los logos contenidos en la Base de Datos

Tabla 7 Clase Entidad Hilo_Identificar

Nombre	Logo
Tipo	Entidad
Atributos	Descripción
nombre	Qstring
direccion	Qstring
categoria	Qstring
relacionAspecto	Qstring
alto	int
ancho	int
Para cada responsabilidad	
Nombres	Logo()
Descripción	Constructor de la clase, se encarga de inicializar el objeto asignando a los atributos de este los pasados por parámetros.
Nombre	Nombre()
Descripción	Retorna el atributo nombre del logo
Nombre	Direccion()
Descripción	Retorna el atributo dirección de localización de los logos.
Nombre	Categoria()
Descripción	Retorna el atributo categoría que define el tipo de programación que transmite la televisora representada por este logo.
Nombre	RelacionAspecto()
Descripción	Resolución original del audiovisual donde estaba contenido el logo al ser detectado para posteriormente almacenarse.
Nombre	Alto()
Descripción	Retorna el atributo altura de la resolución original
Nombre	Ancho()
Descripción	Retorna el atributo ancho de la resolución original

Tabla 8 Clase Entidad Logo

Nombre	I_Plugin_Ejecucion
Tipo	Interfaz

Atributos		Descripción
Para cada responsabilidad		
Nombre	I_Plugin_Ejecucion()	
Descripción	Constructor de la clase.	
Nombre	Ejecutar_Funcionalidad()	
Descripción	Ejecutar una funcionalidad cualquiera sobre un proceso de ejecución.	
Nombre	Ejecutar_Proceso()	
Descripción	Realizar la ejecución del proceso.	
Nombre	Cancelar_Proceso()	
Descripción	Detiene el proceso de monitorización en cualquier instante.	
Nombre	Es_Plugin_Necesario()	
Descripción	Verifica que el plugin se ejecute sobre los parámetros que se están enviando.	
Nombre	Get_Id_Proceso()	
Descripción	Retorna el identificador del proceso necesario para su posterior reconocimiento por parte de la Plataforma.	
Nombre	Get_Nuevo_Objeto()	
Descripción	Devuelve una copia del plugin para que sea almacenado en memoria.	

Tabla 9 Clase I_Plugin_Ejecucion

Nombre	I_Publicador_Ejecucion	
Tipo	Interfaz	
Atributos		Descripción
Para cada responsabilidad		
Nombre	I_Publicador_Ejecucion()	
Descripción	Constructor de la clase.	
Nombre	setInterfaceObserver()	
Descripción	El patrón Observador necesita conocer quién va a observar la clase.	

Tabla 10 Clase I_Publicador_Ejecucion

Nombre	I_Observador_Ejecucion	
Tipo	Interfaz	
Atributos		Descripción
Para cada responsabilidad		
Nombre	I_Observador_Ejecucion()	
Descripción	Constructor de la clase.	

Nombre	Event_Porciento
Descripción	Recibe una notificación con el estado de completamiento porcentual de la monitorización.
Nombre	Event_Error
Descripción	Tratamiento de errores.
Nombre	Event_Termino
Descripción	Finalización del proceso.
Nombre	Event_Variable
Descripción	Variables generadas por el proceso.

Tabla 11 Clase I_Observador_Ejecucion

2.13 Conclusiones parciales

En el capítulo que finaliza se especificaron temas relacionados con el diseño del plugin para la identificación automática de televisoras. Existen algunos elementos destacables:

- El modelo de dominio resultante así como su correspondiente diagrama permitieron alcanzar una mejor comprensión sobre el funcionamiento del futuro sistema.
- Un total de nueve requisitos, seis funcionales y tres no funcionales, fueron definidos y asociados a cuatro casos de uso propios del sistema.
- Se estableció el funcionamiento de la futura aplicación a partir del diagrama de casos de uso y las descripciones los mismos.
- Se identificaron los patrones GRASP y GoF útiles para la aplicación permitiendo facilitar la asignación de responsabilidades logrando un diseño de software que sirva de apoyo a la implementación del sistema.
- Con los diagramas de secuencia y de clases del diseño se esclarecieron las relaciones entre las clases por medio de funciones y se pudo tener una visión estática del sistema.

La investigación puede avanzar de esta manera hacia un estado en el cual se puedan utilizar los elementos expuestos para apoyar la implementación de la solución propuesta así como el proceso de prueba y validación del software al que se enfrentará.

CAPÍTULO 3: Implementación y Pruebas.

3.1 Introducción

En este capítulo se abundará en la implementación del sistema. El trabajo con materiales audiovisuales requiere de un alto consumo de procesamiento por lo que un plugin desarrollado para este tipo de tarea estará basado en programación multihilos. La programación multihilos utilizando las librerías de Qt asegura un eficiente consumo de recursos de hardware, proporcionando además la posibilidad de la realización simultánea de varias tareas.

El flujo de trabajo de implementación detalla cómo los elementos del modelo del diseño se implementan en términos de componentes y representa cómo se organizan en el modelo de despliegue, describiendo los componentes a construir, su organización y dependencias entre los nodos físicos en la que funcionará la aplicación. Se brindará también elementos de interés sobre las pruebas que se aplicaron a la solución y sus resultados con el objetivo de validar el sistema y garantizar su calidad.

3.2 Modelo de implementación

El modelo de implementación está compuesto por un conjunto de subsistemas y componentes que establece la composición física de la implementación del sistema (**Jacobson, y otros, 2004**).

3.2.1 Diagrama de despliegue

Es un tipo de diagrama del UML que muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Se muestra el sistema desde la perspectiva de un programador y se ocupa de la gestión del software; o en otras palabras, se va a mostrar cómo está dividido el software en componentes y las dependencias que hay entre esos componentes. En la figura se muestra el diagrama de despliegue del plugin (**Jacobson, y otros, 2004**).

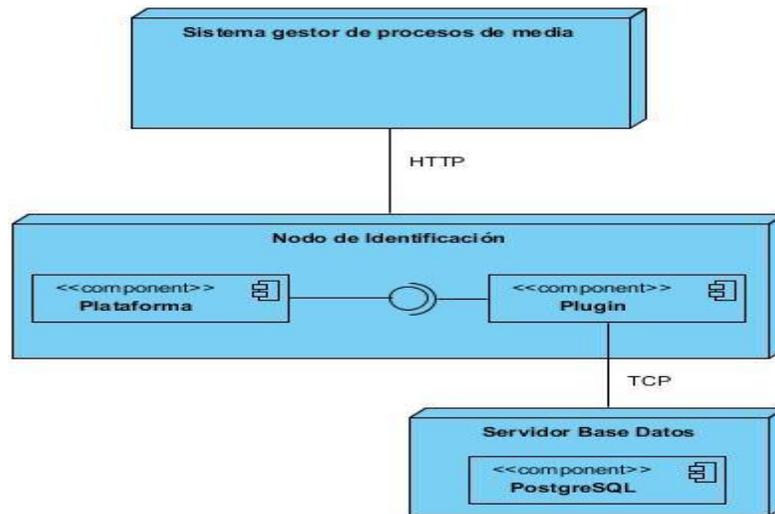


Figura 9 Diagrama de despliegue.

3.2.2 Diagrama de componentes

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en un modelo de diseño. Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, sean estos componentes de código fuente, binarios o ejecutables (Jacobson, y otros, 2004).

3.2.3 Diagrama de componentes de código fuente

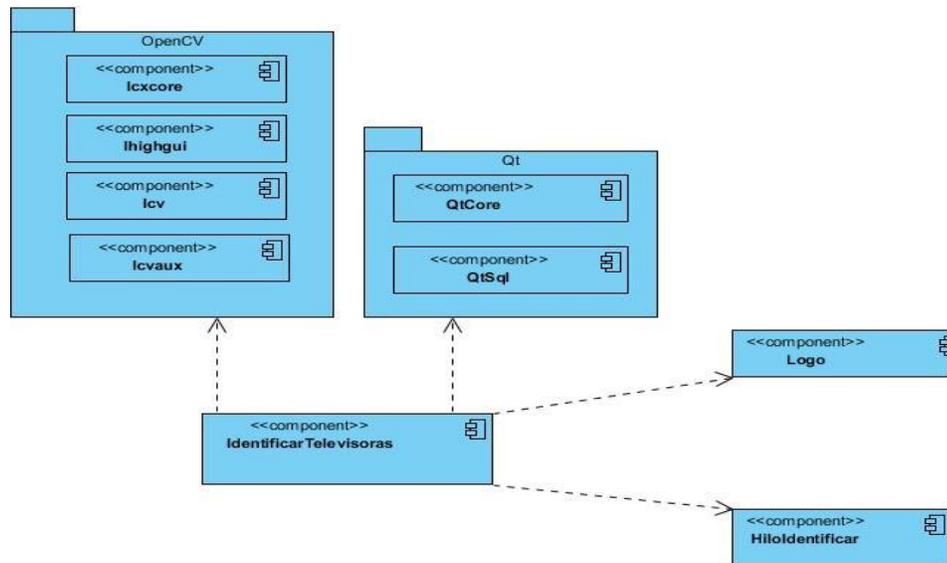


Figura 10 Diagrama de componentes.

3.3 Pruebas de Software

Las pruebas de software son las herramientas con las que se puede conocer la calidad de un producto informático. Identifican posibles errores de implementación, usabilidad o calidad. Son una actividad en la cual un sistema o componente es ejecutado bajo determinadas condiciones o requerimientos, se observan los resultados y posteriormente se registran. Son un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación.

Existen dos técnicas fundamentales en el proceso de pruebas: Caja Blanca y Caja Negra. Cada una posee características particulares:

- **Caja Blanca:** Se analiza la estructura lógica del programa. Se comprueban los caminos lógicos del software proponiendo casos de prueba que examinen que están correctas todas las condiciones y/o bucles para determinar si el estado real coincide con el esperado o afirmado.
- **Caja Negra:** Realiza pruebas de forma que se compruebe que cada función es operativa. Existirá una entrada adecuada y una salida positiva.

Tras un estudio realizado se decidió aplicarle al sistema pruebas funcionales utilizando el método de Caja Negra. Con la realización de las pruebas de caja negra se puede encontrar errores funcionales que puede presentar el software y no han sido detectados por los desarrolladores.

3.3.1 Pruebas de Caja Negra

Se encuentran centradas en los requisitos funcionales del software. Este método de prueba es realizado a nivel de sistema, es decir no se tiene ninguna relación con el código del producto. Un caso de prueba de este tipo incluye la verificación del resultado de la interacción entre el actor y el sistema, que se satisfacen las precondiciones y pos condiciones especificadas por el caso de uso y que se sigue la secuencia de acciones específicas por el caso de uso (**Pressman, 2004**).

3.4 Diseños de casos de pruebas

Los casos de prueba especifican como probar el sistema, incluyendo la entrada o resultados con los que se harán las pruebas y las condiciones en las que se van a probar. Un caso de prueba puede derivarse y por tanto puede seguir la traza de un caso de uso.

3.4.1 Diseños de pruebas de Caja Negra

SC: Monitorizar Media

Escenario	Descripción	Respuesta del Sistema	Flujo Central
EC 1.1 Iniciado el proceso sin errores.	El sistema automático envía la dirección de origen de la media.	El sistema comprueba la validez de la dirección recibida. Comienza la monitorización de la media.	
EC 1.2 La dirección que envía el proceso automático no es válida.	El proceso automático envía la dirección de origen de la media.	El sistema detecta errores al intentar validar la dirección recibida. Notifica al Sistema Automático el mensaje: "La dirección origen no existe o no se tiene permisos de lectura sobre ella". Detiene proceso.	
EC 1.3 Se encuentra un logo durante la monitorización	Búsqueda de un logo en la señal a partir de un algoritmo de identificación de patrones. Se encuentra un logo.	Conectar el sistema a la Base de Datos para buscar posibles coincidencias.	

.			
EC 1.4 No se encuentra logo durante la monitorización	Búsqueda de un logo en la señal a partir de un algoritmo de identificación de patrones. No se encuentra un logo.	El proceso continúa hasta ser detenido por el sistema automático o se completa el proceso de monitorización.	
EC 1.5 Se conecta a la Base de Datos que contiene los logos	El plugin se conecta a la Base de Datos donde se almacenan los logos para realizar las comparaciones entre el logo detectado y los existentes en ella.	Se conecta a la BD	
EC 1.6 No se conecta a la Base de Datos que contiene los logos	El plugin intenta conectarse a la Base de Datos donde se almacenan los logos para realizar las comparaciones entre el logo detectado y los existentes en ella. No es posible.	Notificar al Sistema Automático el mensaje: "No se puede acceder a la Base de Datos"	
EC 1.7 Identifica Televisora a partir de logo detectado	El logo detectado es comparado con los logos presentes en la Base de Datos de logos. Se encuentra una coincidencia.	Se generan variables de interés: instante de aparición, instante de ocultamiento y nombre de la televisora.	
EC 1.8 La señal proviene de un archivo almacenado con anterioridad	La media que se va a monitorizar posee tiempo de duración.	El sistema almacena la duración de la media para emitir un estado de la monitorización.	

Tabla 12 Caso de Prueba-Monitorizar Media

SC: Notificar Porciento

Escenario	Descripción	Respuesta del Sistema	Flujo Central
EC 1.1 El proceso de monitorización almacena una variable duración de la media.	Se utiliza la duración de la media para calcular el avance porcentual que realiza el proceso de monitorización. Se genera la variable por ciento.	Envía al sistema automático una variable que contiene el porciento de ejecución de la tarea.	

Tabla 13 Caso de Prueba-Notificar Porciento

SC: Notificar Evento

Escenario	Descripción	Respuesta del Sistema	Flujo Central
EC 1.1 Se ha identificado una televisora	Es inicializado el proceso de notificación de eventos a partir de la identificación de una televisora.	Se generan y envían a la plataforma las variables Tiempo de Aparición, Tiempo de Ocultamiento y Nombre de la Televisora.	

Tabla 14 Caso de Prueba-Notificar Evento

SC: Detener Proceso

Escenario	Descripción	Respuesta del Sistema	Flujo Central
EC 1.1 El Sistema Automático detiene el proceso de monitorización	El Sistema Automático solicita la interrupción del proceso de forma definitiva.	Se detiene el proceso.	

Tabla 15 Caso de Prueba-Detener Proceso

3.5 Resultados de las pruebas

Se realizó un diseño de caso de prueba para cada Caso de Uso, esto representa un total de cuatro casos de prueba. La primera iteración realizada arrojó 1 no conformidad, la cual se muestra a continuación:

No.	No conformidad	CP	Clasificación
1.	Los datos enviados por el sistema automático son correctos pero el sistema no puede acceder al video.	Monitorizar Media	Alta

Tabla 16 Resultado de la primera iteración de pruebas

Para la segunda iteración se detectaron 2 no conformidades relacionadas con la monitorización de media:

No.	No conformidad	CP	Clasificación
1.	No se conecta a la Base de Datos de logos.	Monitorizar Media	Alta
2.	Emite una alta cantidad de falsos positivos cuando intenta detectar un logo.	Monitorizar Media	Alta

Tabla 17 Resultado de la segunda iteración de pruebas

En la tercera iteración se detectaron 2 no conformidades en el marco de la notificación de eventos:

No.	No conformidad	CP	Clasificación
1.	No percibe los cambios entre televisoras de forma correcta.	Notificar Evento	Alta
2.	Notifica de forma incorrecta el instante de ocultamiento de las televisoras.	Notificar Evento	Media

Tabla 18 Resultado de la tercera iteración de pruebas

En la cuarta iteración no se encontraron no conformidades, esto avaló al plugin desarrollado como una propuesta con calidad para integrar los productos del Departamento Señales Digitales.

3.6 Conclusiones parciales

El estado de la implementación y la calidad del producto fueron analizados:

- Se realizaron diagramas de despliegue y de componentes de implementación, estos brindaron una vista del sistema con una detallada descripción de su estructura.
- Fueron realizadas pruebas de caja negra. Se comprobó el funcionamiento del plugin en cuatro iteraciones de las cuales se arrojaron cinco no conformidades que han sido resueltas.

CONCLUSIONES GENERALES

Con el presente trabajo de diploma se logró dar cumplimiento a las tareas de la investigación propuestas inicialmente, logrando obtener como resultado un plugin para la identificación automática de televisoras e integrable con la Plataforma de Codificación e Indexación. Además, se arribaron a las siguientes conclusiones:

- Se le da respuesta a una necesidad del departamento Señales Digitales con herramientas y tecnologías definidas por sus proyectos.
- Los artefactos generados durante el proceso de desarrollo de software que han sido abordados en la presente investigación proveen de documentación sobre el sistema para futuras actualizaciones y desarrollo de nuevas funcionalidades.
- La integración del plugin con los productos del departamento Señales Digitales permite que estos estén al nivel de otras soluciones foráneas.
- Se desarrolló un plugin que identifica televisoras con un alto grado de efectividad.

RECOMENDACIONES

- Implementar un algoritmo con una técnica que no se vea afectada por los cambios de escala y de esta forma aumentar la fiabilidad del plugin.

REFERENCIAS

1. Acharya, Tinku y Ray, Ajoy k. 2005. ***Image Processing. Principles and Applications.*** Tucson, Arizona : John Wiley & Sons, 2005. 13 978-0-471-71998-4.
2. Acosta Jiménez, Dariel. 2011. ***Componente para la detección de logos en medios audiovisuales.*** La Habana : Universidad de las Ciencias informáticas, 2011. TD-04473-11
3. Alhir, Sinan Si. 1998. ***Que es UML. UML in a Nutshell .*** s.l. : O'Reilly Media, 1998.
4. Bartolome, Antonio. 2008. ***Video Digital y Educacion.*** s.l. : Sintesis, 2008. 9788497565721.
5. Bass, L, Clements, P y Kazman, R. 2003. ***Software Architecture in Practice.*** s.l. : Addison-Wesley, 2003.
6. Bertini, Marco, Del Bimbo, Alberto y Torniai, Carlo. 2005. ***Automatic Video Annotation using Ontologies Extended.*** New York : ACM, 2005. 1-59593-044-2.
7. Cabrera Martínez, Beny. 2011. ***Plataforma de codificación e indexación de video para el Sistema de Captura y Catalogación de Medias.*** La Habana : Universidad de las Ciencias Informáticas, 2011. TD-04371-11.
8. CIVOLUTION. CIVOLUTION. [En línea] CIVOLUTION. [Citado el: 13 de noviembre de 2012.] <http://www.civolution.com/applications/media-intelligence/teletrax-television-monitoring/>.
9. Definición ABC. 2007-2013. Definición ABC. [En línea] Definición ABC, 2007-2013. [Citado el: 16 de noviembre de 2012.] <http://www.definicionabc.com/tecnologia/imagen-digital.php>.
10. DefiniciónABC. 2007-20013. Definición ABC. [En línea] Definición ABC, 2007-20013. [Citado el: 16 de noviembre de 2012.] <http://www.definicionabc.com/comunicacion/canal-de-tv.php>.
11. Douglas, Korry y Douglas, Susan P. 2003. ***PostgreSQL: A Comprehensive Guide to Building, Programming, and Administering PostgreSQL Databases.*** s.l. : Sams Publishing, 2003.
12. García Mateos, Dr Ginés. 2006. Dr Ginés García Mateos. [En línea] 2006. [Citado el: 29 de noviembre de 2012.] <http://dis.um.es/profesores/ginesgm/files/doc/pav/tema6.pdf>.

13. González, Carlos Daniel. 2008. **usabilidadweb.com.ar**. [En línea] **usabilidadweb.com.ar**, 2008. [Citado el: 8 de enero de 2013.] <http://www.usabilidadweb.com.ar/cpp.php>.
14. Gonzalez, Rafael C. y Woods, Richard E. 2002. **Digital Image Processing**. New Jersey : Prentice-Hall, 2002. 0-201-18075-8.
15. Hartung, Frank y Girod, Bernd. 2002. **Watermarking of uncompressed and compressed video**. Elsevier North-Holland : Inc Amsterdam, 2002. 10.1016/S0165-1684(98)00011-5.
16. Hernández Franco, Carlos, Lloret Mauri, Jaime y Viñals Blasco, María José. 2008. **Formación en Inteligencia Artificial y Sistemas Digitales Aplicadas al Control de una Organización de la Comunidad Valenciana**. Valencia : s.n., 2008.
17. **Image Registration by Template Matching Using Normalized Cross-Correlation** . Sarvaiya, J.N. 2009. Kerala : s.n., 2009. 978-0-7695-3915-7.
18. Instituto Nacional de Tecnologías de la comunicación. 2013. **INTECO**. [En línea] **Ministerio de Energía, Industria y Turismo**, 2013. [Citado el: 4 de Febrero de 2013.] http://www.inteco.es/wikiAction/Seguridad/Observatorio/area_juridica_seguridad/Enciclopedia/Articulos_1/Marca_de_agua.
19. ISID. 2006-2009. **ISID**. [En línea] **ISID, 2006-2009**. [Citado el: 13 de noviembre de 2012.] http://www.isid.es/Spanish/product/producto_v01.htm.
20. Jacobson, Ibar, Booch, Grady y Rumbaugh, James. 2004. **El proceso unificado de desarrollo de software**. s.l. : Felix Varela, 2004.
21. Larman, Craig. 2005. **Applying UML and patterns : An Introduction to object - oriented analysis and design and iterative develop** . s.l. : Prentice Hall, 2005.
22. Maldonado, Daniel. 2011. **El Codigo K**. <http://elcodigok.blogspot.com/>. [En línea] 2011. <http://elcodigok.blogspot.com/>.
23. Orúe López, Amalia Beatriz. 2007-2012. **Criptonomicón**. [En línea] 2007-2012. [Citado el: 16 de noviembre de 2012.] <http://www.iec.csic.es/criptonomicon/articulos/expertos88.html>.
24. Pressman, Roger, S. 2004. **Software engineering: a practitioner's approach**. New York : CM, 2004. 978-0-07-337597.

25. Rodríguez Valcarcel, José Antonio. 2006. ***Escudos y marcas tipográficas de editoriales científicas del siglo XX***. México : Investigación bibliotecológica, 2006. 0187-358X.
26. Sierra, Daniel. 2007. **www.slideshare.net**. [En línea] 2007.
<http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>.
27. Sommerville, Ian. 2005. ***Ingeniería de Software***. Madrid : Pearson Educación, 2005. 84-7829-074-5.
28. Sony. 2012. **Sony**. [En línea] Sony Europe Limited, 2012. [Citado el: 8 de enero de 2013.]
<http://www.sony.es/product/dsc-t-series/dsc-t200>.
29. Suárez Pérez, Jean Michael, Fuentes Díaz, Adnan y Becerra Torreira, Yesleny. 2012. ***Arquitectura para Sistema Gestor de Procesos de Media***. La Habana : Sociedad de la Información, 2012.

BIBLIOGRAFÍA

1. Acharya, Tinku y Ray, Ajoy k. 2005. *Image Processing. Principles and Applications*. Tucson, Arizona : John Wiley & Sons, 2005. 13 978-0-471-71998-4.
2. Acosta Jiménez, Dariel. 2011. *Componente para la detección de logos en medios audiovisuales*. La Habana : Universidad de las Ciencias informáticas, 2011. TD-04473-11 .
3. Alhir, Sinan Si. 1998. *Que es UML. UML in a Nutshell* . s.l. : O'Reilly Media, 1998.
4. Bartolome, Antonio. 2008. *Video Digital y Educacion*. s.l. : Sintesis, 2008. 9788497565721.
5. Bass, L, Clements, P y Kazman, R. 2003. *Software Architecture in Practice*. s.l. : Addison-Wesley, 2003.
6. Bertini, Marco, Del Bimbo, Alberto y Torniai, Carlo. 2005. *Automatic Video Annotation using Ontologies Extended*. New York : ACM, 2005. 1-59593-044-2.
7. Cabrera Martínez, Beny. 2011. *Plataforma de codificación e indexación de video para el Sistema de Captura y Catalogación de Medias*. La Habana : Universidad de las Ciencias Informáticas, 2011. TD-04371-11.
8. CIVOLUTION. CIVOLUTION. [En línea] CIVOLUTION. [Citado el: 13 de noviembre de 2012.] <http://www.civolution.com/applications/media-intelligence/teletrax-television-monitoring/>.
9. Definición ABC. 2007-2013. Definición ABC. [En línea] Definición ABC, 2007-2013. [Citado el: 16 de noviembre de 2012.] <http://www.definicionabc.com/tecnologia/imagen-digital.php>.
10. DefiniciónABC. 2007-20013. Definición ABC. [En línea] Definición ABC, 2007-20013. [Citado el: 16 de noviembre de 2012.] <http://www.definicionabc.com/comunicacion/canal-de-tv.php>.
11. Douglas, Korry y Douglas, Susan P. 2003. *PostgreSQL: A Comprehensive Guide to Building, Programming, and Administering PostgreSQL Databases*. s.l. : Sams Publishing, 2003.
12. García Mateos, Dr Ginés. 2006. Dr Ginés García Mateos. [En línea] 2006. [Citado el: 29 de noviembre de 2012.] <http://dis.um.es/profesores/ginesgm/files/doc/pav/tema6.pdf>.

13. González, Carlos Daniel. 2008. [usabilidadweb.com.ar](http://www.usabilidadweb.com.ar). [En línea] [usabilidadweb.com.ar](http://www.usabilidadweb.com.ar), 2008. [Citado el: 8 de enero de 2013.] <http://www.usabilidadweb.com.ar/cpp.php>.
14. Gonzalez, Rafael C. y Woods, Richard E. 2002. *Digital Image Processing*. New Jersey : Prentice-Hall, 2002. 0-201-18075-8.
15. Hartung, Frank y Girod, Bernd. 2002. *Watermarking of uncompressed and compressed video*. Elsevier North-Holland : Inc Amsterdam, 2002. 10.1016/S0165-1684(98)00011-5.
16. Hernández Franco, Carlos, Lloret Mauri, Jaime y Viñals Blasco, María José. 2008. *Formación en Inteligencia Artificial y Sistemas Digitales Aplicadas al Control de una Organización de la Comunidad Valenciana*. Valencia : s.n., 2008.
17. *Image Registration by Template Matching Using Normalized Cross-Correlation* . Sarvaiya, J.N. 2009. Kerala : s.n., 2009. 978-0-7695-3915-7.
18. Instituto Nacional de Tecnologías de la comunicación. 2013. INTECO. [En línea] Ministerio de Energía, Industria y Turismo, 2013. [Citado el: 4 de Febrero de 2013.] http://www.inteco.es/wikiAction/Seguridad/Observatorio/area_juridica_seguridad/Enciclopedia/Articulos_1/Marca_de_agua.
19. ISID. 2006-2009. ISID. [En línea] ISID, 2006-2009. [Citado el: 13 de noviembre de 2012.] http://www.isid.es/Spanish/product/producto_v01.htm.
20. Jacobson, Ibar, Booch, Grady y Rumbaugh, James. 2004. *El proceso unificado de desarrollo de software*. s.l. : Felix Varela, 2004.
21. Keith, J., & Tsatsulin, V. (2002). *Dictionary of video and television technology*. Gulf Professional Publishing.
22. Laganière, R. (2011). *OpenCV 2 Computer Vision Application. Programming Cookbook*. Birmingham, B27 6PA, UK.: Packt Publishing.
23. Larman, Craig. 2005. *Applying UML and patterns : An Introduction to object - oriented analysis and design and iterative develop* . s.l. : Prentice Hall, 2005.
24. Maldonado, Daniel. 2011. El Código K. <http://elcodigok.blogspot.com/>. [En línea] 2011. <http://elcodigok.blogspot.com/>.

25. Orúe López, Amalia Beatriz. 2007-2012. **Criptonomicón**. [En línea] 2007-2012. [Citado el: 16 de noviembre de 2012.]
<http://www.iec.csic.es/criptonomicon/articulos/expertos88.html>.
26. Peña, E. S. (2006). **Segmentación automática de video**. Universidad de Málaga.
27. Pressman, Roger, S. 2004. **Software engineering: a practitioner's approach**. New York : CM, 2004. 978-0-07-337597.
28. Rodríguez Valcarcel, José Antonio. 2006. **Escudos y marcas tipográficas de editoriales científicas del siglo XX**. México : Investigación bibliotecológica, 2006. 0187-358X.
29. Rodríguez Veranes, A. M. (2012). **Componente para localizar regiones de subtítulos en videos**. La Habana: Universidad de las Ciencias Informáticas.
30. Sierra, Daniel. 2007. www.slideshare.net. [En línea] 2007.
<http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>.
31. Sommerville, Ian. 2005. **Ingeniería de Software**. Madrid : Pearson Educación, 2005. 84-7829-074-5.
32. Sony. 2012. **Sony**. [En línea] Sony Europe Limited, 2012. [Citado el: 8 de enero de 2013.] <http://www.sony.es/product/dsc-t-series/dsc-t200>.
33. Suárez Pérez, Jean Michael, Fuentes Díaz, Adnan y Becerra Torreira, Yesleny. 2012. **Arquitectura para Sistema Gestor de Procesos de Media**. La Habana : Sociedad de la Información, 2012.