Universidad de las Ciencias Informáticas

Facultad 7

Título: Sistema para la Detección y Extracción de Textos en Videos Digitales.

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

Autores: Yoel Rivera Suarez.
Fernando Echemendia Tourt.

Tutores: MSc. Eduardo Solís Céspedes.
Ing. Heliodoro Rodríguez Milian.

Ciudad de la Habana, Mayo de 2008.

DECLARACIÓN DE AUTORÍA

Declaro que somos los únicos autores de este trabajo y autorizamos al Grupo de Procesamiento de Imágenes de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 31 días del mes de mayo del año 2008.

Yoel Rivera Suárez.

_____

MSc. Eduardo Solís Céspedes.

_____

Fernando Echemendia Tourt

_____

Ing. Heliodoro Rodríguez Milian

_____

DATOS DE CONTACTOS

MSc. Eduardo Solís Céspedes (email: esolis@uci.cu).

Profesor graduado de Licenciatura en Radioquímica y Máster en Informática Aplicada. Ha impartido las asignaturas de Física I, Física II y Práctica Profesional. Es profesor de la facultad 7, jefe de proyecto dentro del Grupo de Procesamiento de Imágenes (GPI) de la Universidad de las Ciencias Informáticas (UCI), actualmente se desempeña como vice decano de producción de la facultad.

Ing. Heliodoro Rodríguez (email: hrodriguezm@uci.cu).

Profesor graduado de Ingeniero Informático. Es profesor de la facultad 7, jefe de proyecto dentro del Grupo de Procesamiento de Imágenes (GPI) de la Universidad de las Ciencias Informáticas (UCI).

AGRADECIMIENTOS

Agradezco a todas las personas que de una forma u otra han contribuido al desarrollo del presente trabajo de diploma. Primeramente a la Revolución por darme la posibilidad y derecho de hacerme un profesional. En segundo lugar a mis padres y familiares por apoyarme en todo momento y tenerlos siempre a mi lado, por sus consejos y alientos en los momentos mas decisivos de mi vida.

A nuestros tutores Eduardo y Helio, y a la profesora Pura por su apoyo total, sirviendo siempre de guía en cada idea de desarrollo. A Dannier y Yurisnel por haberme dado su ayuda incondicional, contribuyendo en el desarrollo del trabajo.

Yoel.

Agradecer primeramente a la Revolución por permitirme crecer como persona. A Eduardo y Helio por su invaluable ayuda. A la profesora Pura por su paciencia y dedicación. A mis compañeros de aula por su amistad y a todas las personas que hicieron posible la realización de este trabajo.

Fernando.

DEDICATORIA

A mis padres,mis abuelos y a Yici, por haber aguardado a mi lado en mis momentos buenos, difíciles y malos con tanto amor, cariño y comprensión; permitiéndome lograr mi sueño, convertirme en ingeniero informático.A mi familia y amigos por la fe y la confianza.

Yoel.

A mis padres, en especial a mi madre por su ejemplo y amor infinito. A mi hermanita por quererme tanto. A toda mi familia, en especial a mi prima por su apoyo incondicional. A mis amigos por estar siempre en los momentos dificiles. A todos mis profesores por su esfuerzo y dedicación.

Fernando.

RESUMEN

En la actualidad existe la necesidad de realizar búsquedas en videos digitales por su contenido. De forma tradicional, se crean catálogos hechos manualmente, donde se almacenan las extensiones, la duración, el título, entre otras características descriptivas que no logran sintetizar el contenido de los mismos. Todo ello ocasiona que, muchas veces la eficiencia de las búsquedas no sea la deseada.

El presente trabajo propone desarrollar un sistema que sea capaz de detectar, extraer y procesar el texto artificial contenido en fotogramas de videos digitales, con el propósito de usarlo posteriormente; en la búsqueda en videos por contenido. Esta aplicación es robusta ante la presencia de textos horizontales, que posean un contraste relativamente alto y es invariante a la escala del texto. Para su realización fue utilizado el lenguaje de programación c# 2.0, así como la metodología RUP para regir todo el proceso de ingeniería.

Este sistema informático cuenta con los elementos requeridos en materia de procesamiento de imágenes digitales. Hace uso además, de novedosas técnicas en el campo de reconocimiento de patrones para detectar y extraer el texto contenido en videos digitales.

PALABRAS CLAVES

Detección de texto, detección de bordes, fotogramas, morfología matemática, proyecciones.

TABLA DE CONTENIDO

INTRODUCCIÓN

El creciente desarrollo de las Tecnologías de la Informática y las Comunicaciones (TIC) le ha proporcionado al hombre una de las herramientas más potentes con las que ha podido contar durante toda la historia de la humanidad. Las relaciones humanas, comerciales y de negocios están soportadas hoy, en su mayoría, sobre la existencia de computadoras y redes de computadoras que nos permiten intercambiar una gran cantidad de información. Este flujo de datos ha dado lugar al auge de los contenidos multimedia (imágenes, vídeos y audio).

En estos últimos años, el ámbito multimedia se ha convertido en uno de los campos más prometedores y de constante evolución. Ello se debe tanto por las mejoras de dispositivos físicos como por la evolución de técnicas y algoritmia. De manera creciente las compañías están entendiendo que dado el rápido incremento en el número de contenido multimedia disponible en la Internet, la forma para aprovechar mejor esa oportunidad y obtener beneficios de ella es convertirse en un almacén que facilite el acceso a esta inmensa cantidad de contenido.

Con archivos multimedia, en especial videos digitales que tengan la posibilidad de hacer una búsqueda mediante palabras claves, se les hace a los usuarios mucho más sencillo encontrar el contenido que pueda adecuarse a sus necesidades específicas. Así como, extiende la posibilidad de apuntar a citas específicas, frases habladas y afirmaciones que pueden tener infinitas aplicaciones en centenares de campos, desde la publicidad, las leyes hasta en lo científico.

La tecnología de búsqueda de video se está convirtiendo rápidamente en algo imprescindible en el estilo de vida digital actual, ya que más consumidores tienen acceso a conexiones Internet de alta velocidad y consumen más contenido de audio y video que nunca antes.

Dentro de este marco tecnológico, la digitalización y tratamiento de video ha irrumpido con gran fuerza. Sin embargo, no es hasta hace poco cuando se han superado las barreras de complejidad computacional que supone su manipulación.

El texto, contenido en los fotogramas, es uno de los elementos que más información brinda sobre el contenido del video. En una secuencia de imágenes se pueden encontrar dos tipos de texto embebido: captions, que forman texto superpuesto artificialmente sobre la secuencia, y textos vinculados naturalmente a la escena como señales o anuncios. Los textos propios de la escena pueden

proporcionar información semántica de la secuencia, mientras que la información brindada por los captions está más enfocada al contenido, por lo tanto, ambos son útiles a la hora de indexar el video.

Actualmente la información del contenido de los videos se procesa de forma manual y es archivada para posteriores búsquedas. Este proceso de anotación manual de videos tiene varios inconvenientes: es costoso, tedioso, necesita mucho tiempo de trabajo, además de estar sujeto a ambigüedades y errores humanos; de ahí que del análisis de la problemática expuesta surja el planteamiento del siguiente **problema**: ¿Cómo detectar y extraer el texto contenido en videos digitales de manera eficiente?

Con vistas a dar solución al problema planteado, quedan definidos como **objeto de estudio** los procesos de detección y extracción de texto contenido en imágenes. A raíz de lo cual, el **campo de acción** se enmarca en los procesos de detección y extracción de texto presente en imágenes contenidas en videos digitales.

El **objetivo general de la investigación** es desarrollar un sistema que detecte y extraiga el texto contenido en videos digitales.

En la presente investigación quedaron definidas una serie de **tareas**:

- Implementar la forma más eficiente de descomponer los videos digitales en fotogramas.
- Analizar, en el campo de procesamiento de imágenes digitales, los métodos existentes para la detección y extracción de texto contenido en imágenes.
- Implementar la metodología más eficiente y viable para la detección y extracción de texto contenido en videos digitales.
- Realizar un estudio de varios OCR (Optical Character Recognition) para utilizar en la solución el de mejores resultados.

Para el cumplimiento de las tareas antes mencionadas, se emplearon los siguientes **métodos científicos**:

Métodos Teóricos.

Modelación: Este método se considera de gran importancia para la confección del sistema puesto que para su realización se ha utilizado como metodología RUP[1], con la cual se hace necesaria la creación de varios artefactos en los diferentes modelos que se construyen, permitiendo esto una reproducción amplia de la realidad.

Analítico – Sintético: Ha permitido avanzar en la solución del sistema. Es conocer las teorías y documentos que ocupan el objetivo de investigación, extrayendo los aspectos más importantes relacionados con el objeto de estudio.

Histórico – Lógico: Ayuda a la comprensión de la evolución en el mundo de los sistemas para la búsqueda y detección de textos en videos a lo largo de la historia.

Métodos Empíricos.

Observación: Este método es de suma importancia puesto que permite observar mediante el registro visual lo que ocurre en la situación real que se analiza.

Entrevista: Mediante la aplicación de este método se puede saber los criterios del cliente respecto al sistema, además, mediante él surgen nuevos requerimientos para el software.

El trabajo consta de cinco capítulos donde se realiza un estudio crítico y descriptivo de las tecnologías seleccionadas, un estudio comparativo para la selección de los métodos óptimos de procesado y detección de texto. Se plantearán las características del sistema comenzando por una vista de los procesos del negocio existentes; se diseñará el sistema propuesto y luego se implementará cumpliendo así con los principales flujos del ciclo de desarrollo.

En el Capítulo 1 se tratan los temas relacionados con la fundamentación teórica que justifica la investigación, se analiza el estado actual del tema a tratar a nivel nacional e internacional así como las nuevas tendencias y las tecnologías usadas en la solución, se fundamenta la metodología usada y se abordan otros temas de interés.

En el Capítulo 2 se expone todo lo referente al algoritmo que se utiliza en el sistema, para la detección del texto contenido en cada fotograma. Además se hace un estudio sobre su eficiencia, exponiendo resultados experimentales concretos.

[1] Rational Unified Process

En el Capítulo 3 se abordan las características del sistema propuesto, se describe el objeto de automatización y la propuesta de sistema, se analiza el Modelo de Negocio que incluye la presentación de los actores y trabajadores, los diagramas de casos de uso del negocio y los de actividades, se presenta además el diagrama de clases correspondiente al Modelo de Objetos y se describen los requerimientos del sistema que constituyen la base para la posterior realización del Modelo de Casos de Uso.

El Capítulo 4 está dedicado a la arquitectura y el diseño del sistema a través de los modelos arquitectónicos y de diseño donde se detallan las características de la arquitectura empleada y se definen las clases del diseño y sus relaciones respectivamente.

En el Capítulo 5 se tratarán los artefactos generados por el Modelo de Implementación, esencialmente el diagrama de componentes donde se hace un profundo análisis de cada componente y la relación entre ellos.

Los Anexos incorporados al final del documento contienen una muestra de los resultados obtenidos por el método de detección de texto propuesto.

En el Glosario de Términos se definen claramente los términos relacionados con el tema que se presenta u otro tema que pueden ser de difícil comprensión.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

En este capítulo se abordan los temas del proceso de detección y extracción de texto en videos digitales, para ubicar el contexto en el que puede enmarcarse el sistema propuesto por la presente investigación. Se hace un estudio del estado del arte sobre los sistemas automatizados y métodos existentes en la actualidad, en el ámbito nacional e internacional, que abordan los procedimientos de extracción del texto. Se presentan las principales tendencias en el mundo de las tecnologías para darle solución a problemas similares al tratado en esta investigación. También se analizan las principales técnicas, tecnologías, herramientas y metodologías utilizadas para desarrollar el sistema.

1.1 Imagen digital.

Una imagen digital es una matriz de MxN elementos que se obtienen mediante la discretización 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. Cada elemento de la matriz se denomina pixel[2] (Gonzalez and Woods 2002) .

1.1.1 Imagen digital en escala de grises

Es una matriz de MxN elementos numéricos, siendo este número la intensidad luminosa en un determinado punto o pixel (Molina 1998).

1.1.2 Imagen digital a colores RGB

Las imágenes digitales en el modelo de color RGB[3] (Rojo, Verde, Azul) están formadas por 3 matrices de MxN elementos numéricos, siendo este número la intensidad luminosa en cada una de las bandas espectrales de cada punto o pixel (Molina 1998). A diferencia de las imágenes en escala de grises, las imágenes a color requieren de la combinación de las 3 bandas de color, para representar el color de un pixel.

[2] píxel (del inglés *picture element*, "elemento de la imagen"), menor unidad en la que se descompone una imagen digital.
[3] Espacio de color RGB (Red Green Blue, del inglés), sistema de síntesis aditiva basado en rojo, verde y azul.

1.2 Video digital.

El vídeo digital es un tipo de sistema de grabación de vídeo que funciona usando una representación digital de la señal de vídeo. 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 (Bimbo 2005). Está compuesto por fotogramas, que son cada una de las imágenes individuales de una secuencia o animación, por tomas, que son las secuencias de imágenes realizadas en un mismo plano y por escenas, que representan un conjunto de tomas relacionadas entre si.

1.3 Texto Artificial en videos digitales.

En una secuencia de video se pueden encontrar dos tipos de texto embebido: *captions,* y los textos *propios* de la escena, que aunque pueden proporcionar información semántica de la secuencia, la proporcionada por los *captions* es considerablemente mayor y, por lo tanto, más útil a la hora de indexar el video (Cortés 2005). Los c*aptions* son un conjunto de caracteres alfanuméricos que se encuentran superpuestos o incrustados en los fotogramas de videos digitales. Representan una versión textual de diálogos o narraciones que se producen en las escenas o simplemente brindan información. Pueden ser generados en tiempo real o en procesos posteriores a la grabación y generalmente se presentan de forma horizontal y con un alto contraste en relación al fondo del fotograma.

Figura 1 Fotograma con texto artificial.

1.4 Métodos de detección y extracción de texto en videos digitales.

La detección y extracción de texto en videos digitales es un campo de investigación relativamente joven y en constante desarrollo. No es hasta hace algunos años que se supera la barrera de complejidad computacional que supone la manipulación de archivos de video, propiciando esto la proliferación de novedosas técnicas en este marco tecnológico.

Actualmente se pueden identificar dos tendencias fundamentales en las técnicas o métodos vinculados a la detección de texto artificial contenido en fotogramas de videos en el dominio no compreso. Estos son los métodos basados en detección de bordes y los basados en análisis de textura.

1.4.1 Métodos de detección basados en bordes.

Los algoritmos basados en detección de bordes explotan características bien definidas en las líneas de texto artificial. Los captions presentan generalmente alineación horizontal en los fotogramas, con una alta densidad de trazos verticales y los caracteres de una misma línea de texto o de una palabra son del mismo tamaño y se encuentran homogéneamente espaciados. Además de poseer un alto contraste en relación al fondo del fotograma.

Entre las metodologías más representativas de este grupo se puede mencionar la propuesta por Anthimopoulos, Gatos y Pratikakis (Anthimopoulos, Gatos and Pratikakis 2006). Primeramente se realiza una detección de bordes sobre el fotograma original llevado a escala de grises utilizando el operador de Canny. La detección de bordes es esencialmente una operación de detección de cambios de intensidad locales significativos (Molina 1998). Generalmente se emplean operadores locales basados en aproximaciones discretas de la primera derivada de los niveles de grises de la imagen; como son el operador de Sobel, Robert y Prewitt.

Después de realizada la detección de bordes, se aplica una dilatación por un elemento estructurante de 5x21 para conectar contornos de caracteres de una misma línea de texto. La dilatación es la transformación morfológica que combina dos vectores utilizando la suma (Molina 1998). Formalmente se define por la siguiente ecuación.

$$A \oplus \mathrm{B} = \{ \mathrm{p} \in E^2 : \mathrm{p} = \mathrm{a} + \mathrm{b}, \mathrm{a} \in \mathrm{A} \text{ y } \mathrm{b} \in \mathrm{B} \}$$

Donde A es el conjunto de pixeles de la imagen, B es el conjunto de pixeles del elemento estructurante, $\oplus$ simboliza la operación de dilatación, y E^2 es el espacio discreto de todos los posibles puntos (Acharya and Ray 2005).

Figura 2 a) Fotograma con texto artificial b) Mapa de bordes c) Resultado de la dilatación.

Posteriormente se realiza un análisis de componentes conexas para localizar las cajas de texto candidatas. Consiste en realizar el etiquetado y agrupamiento de elementos que pertenecen a un mismo objeto en la imagen. Las cajas de texto candidatas son filtradas por sus características geométricas y son descartadas si no cumplen ciertas restricciones en cuanto a alto, ancho y relación ancho alto. De las cajas de texto que quedan del proceso anterior se extraen sus proyecciones horizontales quedando estas divididas en regiones más pequeñas y pasando nuevamente por el paso de filtrado por características geométricas. La proyección horizontal (Acharya and Ray 2005) de una imagen se define como:

$$HI(i) = \sum_{j=1}^{n} I(i,j)$$

Donde $HI(i)$ es la proyección horizontal en una imagen $I(i,j)$ de tamaño nxm.

Figura 3 a) Análisis de componentes conexas b) Cajas de texto candidatas c) Cajas de texto finales.

Este método basado en bordes es eficaz en la detección de textos de diferentes tamaños. No es sensible a la aparición de caracteres de diferente color y brinda buenos resultados ante líneas de texto complejas. Su principal limitante es el alto contraste que debe presentar el texto artificial para ser detectado correctamente.

1.4.2 Métodos de detección basados en análisis de textura.

Los métodos de detección basados en análisis de textura manipulan el texto artificial como un tipo de textura bien definida y diferenciable del resto de la imagen. Los captions poseen cierta información de frecuencia, orientación y cohesión espacial que los hace identificables mediante la segmentación por textura.

Aunque no existe una definición formal de textura de un objeto, este descriptor proporciona medidas de propiedades como suavidad, regularidad y densidad (Molina 1998). Una de las aproximaciones más utilizadas en la descripción por textura es la estadística. Esta se basa en el cálculo de estadígrafos como entropía, energía y desviación estándar sobre el histograma de la imagen.

En el método propuesto por Wu, Manmatha y Riseman (Wu, Manmatha and Riseman 1997) primeramente se realizan tres suavizados de la imagen con un filtro Gausiano 2D con σ = $(1,\sqrt{2},2)$ respectivamente.

El filtro de alisamiento Gausiano 2D, $G(x, y)$ viene dado por la siguiente ecuación.

$$G(x, y) = c \exp\left[-\frac{x^2+y^2}{2\sigma^2}\right]$$

Donde x e *y* son las coordenadas de la imagen, σ es la desviación típica de la distribución de probabilidad y *c* es una constante de normalización (Molina 1998).

Luego de realizar una transformación no lineal a las imágenes suavizadas se cuantifica la energía de cada una.

La energía E está definida por:

$$E = \sum_{i=1}^{n} \sum_{j=0}^{n} [P_{i,j}]$$

Donde $P_{i,j}$ es la matriz de co-ocurrencia (Acharya and Ray 2005).

Luego se clasifican los pixeles de las distintas imágenes teniendo en cuenta los niveles de energía antes calculados para ser clusterizados con un algoritmo K means con K=3. Quedando etiquetadas en negro las posibles regiones de texto.

Figura 4 a) Imagen original b) Resultado de la segmentación por textura c) Regiones de texto.

Posteriormente se realiza una detección de bordes en las regiones de texto encontradas. Se realiza un análisis de componentes conexas para generar las cajas de textos candidatas. Se aplican restricciones geométricas para eliminar las cajas de texto que no contienen realmente texto.

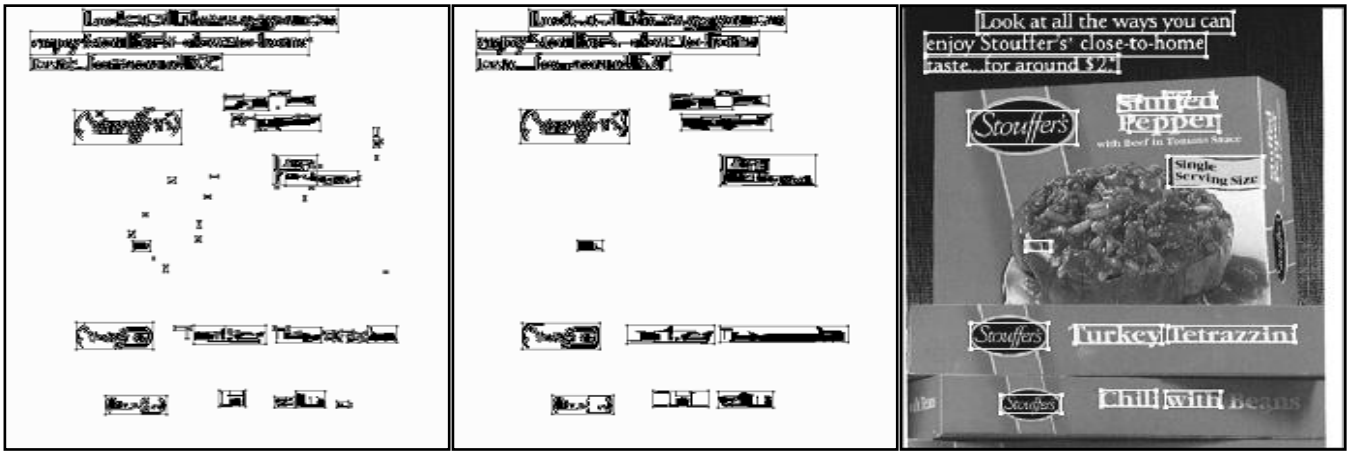


Figura 5 Cajas de texto candidatas b) Cajas de texto refinadas c) Cajas de texto imagen original.

Los métodos de detección basados en análisis de textura son efectivos ante fondos complejos. No son sensibles a textos de diferente tamaño y color. La principal limitante que se puede señalar es el alto costo computacional que implica la segmentación por textura.

1.5 Principales herramientas y tecnologías utilizadas.

1.5.1 Lenguaje de Programación.

Visual C# .NET 2.0.

El lenguaje C# es puramente orientado a objetos, desarrollado por la compañía Microsoft, que posee una constante actualización, sobre el cual se pueden implementar diversas tecnologías muy modernas y utilizadas en la actualidad. Es el denominado lenguaje estrella de la plataforma .NET; ha sido diseñado para ser robusto, moderno y sencillo.

El mismo funciona bajo un entorno de recolección automática de la memoria, lo que aumenta la productividad del desarrollador, brinda un marco de ejecución administrada que permite la optimización y la ejecución segura del código. Se destaca, además, la existencia de tipos genéricos, indexadores, delegados, eventos, clases parciales, estructuras e interfaces, entre otros rasgos.

Es un lenguaje excelente para la utilización de patrones de diseño y metodologías ágiles. Permite la utilización de muy buenas herramientas para el control de versiones como Visual Source Safe, cuando se programa con el Entorno Integrado de Desarrollo (IDE) Microsoft Visual Studio 2005 Team Suite (Microsoft 2008); el cual es muy popular en la comunidad de desarrolladores.

La principal desventaja de utilizar este lenguaje es que las plataformas de desarrollo en las que se lleva a cabo su utilización son privadas. Problema que se puede evitar adoptando una estrategia de migración hacia la plataforma libre Mono.

1.5.2 Lenguaje de Modelado.

UML 2.1

Lenguaje Unificado de Modelado (UML) (IngenieroSoftware n.d.) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software de forma concisa, comprensiva y escalable.

UML ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Cabe destacar que UML es un lenguaje para especificar y no un método.

Es utilizado para definir los componentes y artefactos en un sistema. Además, concentra las mejores prácticas de la comunidad y ha sido adoptado masivamente por la industria. En específico el UML 2.1, que es el que se utilizará en la realización de este sistema, posee mejoras en cuanto a la variedad de diagramas, así como la disposición de nuevos artilugios en los diagramas ya existentes en UML 1.x.

1.5.3 Metodología de Desarrollo

RUP

Proceso Unificado de Desarrollo (RUP) (Molpeceres 2002), constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos. RUP brinda una forma disciplinada para la asignación de las tareas y responsabilidades (quién hace qué, cuándo y cómo). Sus principales características son:

- ✓ Iterativo e incremental: A medida que avanza el proceso de desarrollo se producen versiones incrementales, las cuales se acercan cada vez más al producto terminado.
- ✓ Guiado por los casos de uso: Los casos de uso son los que indican como debe actuar el sistema con el usuario final o con otro sistema para conseguir su objetivo.
- ✓ Centrado en la arquitectura: Los modelos son proyecciones del análisis y el diseño lo cual constituye la arquitectura del producto a desarrollar.

Además está compuesto por cuatro fases que son: Inicio, Elaboración, Construcción y Transición, cada una de ellas compuesta de una o varias iteraciones. A su vez está compuesto por nueve Flujos de Trabajo de Ingeniería: Modelamiento del negocio, Requerimientos, Análisis y diseño, Implementación, Pruebas y Despliegue, Administración de cambio y configuración, Administración de proyecto y Entorno. En cada una de sus fases se emplean todos los flujos de trabajo pero con diferente énfasis, aunque con el mismo objetivo, obtener artefactos de calidad óptima realizando un esfuerzo mínimo.

1.5.4 Herramienta CASE

Enterprise Architect 7.0

Enterprise Architect (EA) es una herramienta CASE[4] que combina el poder de la última especificación UML 2.1 con alto rendimiento, interfaz intuitiva para traer modelado avanzado al escritorio y para el equipo completo de desarrollo e implementación. Posee ventajas tecnológicas con respecto a otras herramientas similares y su costo es relativamente bajo.

Además puede equipar a un equipo entero, incluyendo analistas, evaluadores, administradores de proyectos, personal del control de calidad, equipo de desarrollo y más, por una fracción del costo de algunos productos competitivos. Es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad, aspecto de vital importancia.

EA se encarga de administrar la complejidad con herramientas para rastrear las dependencias, brinda soporte para modelos muy grandes, control de versiones con proveedores CVS[5], SCC, subversion y TFS. Además, permite generar la documentación y posee herramientas de reporte con un editor de

[4] *Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador.

[5] Concurrent Versions System, es una aplicación informática que implementa un sistema de control de versiones.

plantilla completo WYSIWYG (Sánchez 2007). Soporta generación e ingeniería inversa de código fuente para muchos lenguajes populares, incluyendo C++, C#, Java, Delphi, VB.Net, Visual Basic, PHP y ActionScript por solo mencionar algunos. También posee Add-ins gratis para CORBA[6] y Python disponibles. Soporta además transformaciones de arquitectura avanzada dirigida por modelos usando plantillas de transformaciones de desarrollo y fáciles de usar. Exporta modelos de otras herramientas case mediante XML desde 1.1 hasta 2.1.

1.5.5 Plataforma de Desarrollo

Microsoft Visual Studio 2005

Visual Studio .NET es un conjunto completo de herramientas de desarrollo para la construcción de aplicaciones Web ASP, servicios Web XML, aplicaciones para escritorio, archivos DLL[7], aplicaciones de consola y aplicaciones móviles.

Ofrece algunas características exclusivas de alta productividad, tales como: diseñadores visuales de Web Forms y Windows Forms, esquemas XML y datos. Un depurador de varios lenguajes que alterna sin problemas entre códigos escritos en lenguajes diferentes. Proporciona una estrecha integración con .NET Framework; así como una ayuda dinámica, que proporciona completamiento contextual continuo mientras se escribe; muestra una lista de tareas, los errores del compilador y las tareas pendientes. Presenta características de diseño de arquitecturas como la integración con Visio y un explorador de servidores para obtener acceso visual a bases de datos, servicios de Windows, contadores de rendimiento y componentes de aplicaciones del lado del servidor.

Visual Studio es actualmente el IDE[8] por excelencia para la mayoría de los desarrolladores, plataforma solo comparada en algunos aspectos con el NetBeans y el Eclipse ambos para el lenguaje Java, los antes mencionados no cumplen con todas las características vistas. Todas estas particularidades hacen del Visual Studio la mejor y más práctica herramienta para desarrollar nuestra aplicación.

[6] *Common Object Request Broker Architecture*, arquitectura común de intermediarios en peticiones a objetos, es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos.
[7] Dynamic Link Library, Biblioteca de vínculos dinámicos.
[8] Entorno Integrado de Desarrollo.

1.5.6 Control de Versiones

Microsoft Visual Source Safe 2005

Microsoft Visual SourceSafe es un sistema de control de versiones en el nivel de archivos, que permite a muchos tipos de organizaciones trabajar en distintas versiones de un proyecto al mismo tiempo. Esta funcionalidad es especialmente útil en un entorno de desarrollo de software, donde se utiliza para mantener versiones de código paralelas (MSDN n.d.).

Sus características más ventajosas residen en la integración que presenta con Microsoft Visual Studio 2005, lo cual incluye la posibilidad de obtener documentación sobre el mismo en la MSDN de Microsoft; además de su facilidad de uso, dado que la experiencia de usuario que posee es muy rica para sistemas de este tipo. Otras características fundamentales que se podrían mencionar del sistema están dadas por la ya mencionada codificación concurrente, registrar un historial de cambios en los ficheros y la administración del código fuente.

1.5.7 Herramienta Auxiliar

Mono Migration Analyzer (MoMA)

Es una herramienta que ayuda a identificar diferentes problemas que se tengan al portar las aplicaciones del Framework .NET al Framework de Mono. Sin duda alguna, MoMA ayuda a detectar problemas potenciales, que se pueden tener ante la posibilidad de ejecución en entornos UNIX; lo cual es uno de los requerimientos no funcionales del sistema propuesto.

Es por esto que la realización de pruebas de compatibilidad se han hecho con esta novedosa herramienta desarrollada por los miembros del proyecto MONO, la alternativa para la plataforma UNIX de Microsoft .NET. Los resultados obtenidos de la realización de estas pruebas de compatibilidad se detallan en el epígrafe 5.3 del Capítulo 5.

1.6 Reconocimiento Óptico de Caracteres (OCR).

El software de reconocimiento óptico de caracteres, abreviado habitualmente como OCR (Optical Character Recognition), extrae de una imagen los caracteres que componen un texto para almacenarlos en un formato con el cual puedan interactuar programas de edición de texto (Sazaklis 2002).

Mientras que en una imagen los caracteres se describen indicando cada uno de los puntos que los forman, al convertirlos a un formato de texto (por ejemplo ASCII o Unicode), pasan a estar descritos por un solo número, por lo que se produce una reducción significativa del espacio en memoria que ocupan.

A partir de ahí el texto es reconocido como texto, de modo que se pueden buscar en él cadenas de caracteres, exportar el texto a un editor de textos, o a otras aplicaciones, etc.

Se podrían mencionar numerosos softwares de este tipo, de gran eficacia como por ejemplo:

- ABBYY® FineReader

Producto propietario creado por ABBYY Software, proveedor líder mundial de software OCR, ICR y lingüístico. ABBYY FineReader (ABBYY Software 2005) reconoce documentos mono y plurilingües (por ejemplo inglés–francés). Cuando se está reconociendo documentos en inglés o alemán, también se puede utilizar los idiomas correspondientes con diccionarios especializados. Además de los diccionarios médicos y jurídicos, estos idiomas también incluyen diccionarios generales de ABBYY FineReader.

Además permite especificar la velocidad y la calidad del proceso de reconocimiento. Hay dos modos de reconocimiento disponibles:

- ✓ **Exhaustivo**

  En este modo, ABBYY FineReader analiza y reconoce tanto documentos sencillos como complejos, en especial documentos con texto sobre un fondo de color o trama y documentos con tablas complejas (incluidas tablas con líneas de cuadrícula blancas y celdas de colores).

- ✓ **Rápido**

  Es el modo recomendado para procesar muchos documentos con distribuciones sencillas y buena calidad de imagen.

- ✓ **Otras prestaciones**

  El software es capaz de realizar reconocimiento de códigos de barras, orientación del texto, reconocimiento en segundo plano y reconocimiento con moldeado.

- LEADTOOLS Document Imaging Suite (2007)

Aplicación propietaria desarrollada por LEAD Technologies, empresa de más de 10 años de experiencia en este campo. El motor OCR de LEADTOOLS soporta diccionarios personalizables para reconocer palabras que pueden existir solamente en los documentos que están siendo reconocidos. Otras nuevas características del motor OCR son el soporte para alrededor de 100 idiomas, opciones de salida de documentos como los márgenes del documento, opciones de párrafo y más, así como nuevos formatos de salida.

Se soportan tres motores OCR de reconocimiento especializado:

Motor MOR

- ✓ Soporta 114 idiomas
- ✓ Soporta hasta 500 zonas en una imagen
- ✓ Soporta entrenamiento del caracter para alcanzar una precisión mejorada
- ✓ Proporciona ajustes de exactitud y velocidad de 3 niveles de página incluyendo exacto, equilibrado y rápido
- ✓ Proporciona corrección basada en el subsistema de comprobación

Motor MTX (MText)

- ✓ El más rápido de los motores OCR seleccionables.
- ✓ Soporta 12 idiomas.
- ✓ Soporta hasta 64 zonas en una imagen.
- ✓ Proporciona ajustes de exactitud y velocidad de 2 niveles de página incluyendo exacto, equilibrado y rápido.
- ✓ Proporciona corrección basada en el subsistema de comprobación.

Motor FireWorX

- ✓ Optimizado para la velocidad.
- ✓ Soporte para 54 idiomas.
- ✓ Soporta hasta 2.500 zonas en una imagen.

- ✓ Soporta entrenamiento del caracter para alcanzar una precisión mejorada.

- Office Document Imaging (Microsoft 2008).

Creado por Microsoft Corporation e incluido en su paquete propietario Office 2007. Permite digitalizar documentos de una o varias páginas. Por ejemplo, se pueden digitalizar documentos impresos para archivar y reciclar las copias impresas.

Realiza el reconocimiento óptico de caracteres (OCR) en un documento digitalizado o un fax. Por ejemplo, tras el reconocimiento del texto, se puede buscar un texto específico o copiar el texto en otro programa.

Copia además, texto e imágenes de un documento digitalizado y las pega en cualquier programa de Office. Además, exporta el texto y las imágenes a Microsoft Word o una hoja de cálculo de Microsoft Excel.

Teniendo en cuenta que el desarrollo de el sistema se realizará en Microsoft .NET FrameWork, se decidió utilizar LEADTOOLS Document Imaging Suite, pues la potencia de LEADTOOLS combinada con la del Microsoft .NET FrameWork permite a los desarrolladores añadir fácilmente en sus aplicaciones .NET un tratamiento avanzado de imágenes, incluyendo diversos formatos de imagen, compresión, procesamiento de imágenes, visualización, anotaciones, OCR y escaneado.

LEADTOOLS Raster Imaging Pro para .NET está 100% escrito en código supervisado (Managed Code). Además extiende la funcionalidad de GDI+ y del .Net FrameWork, superando las limitaciones y el rendimiento que tienen las clases nativas de .Net para el manejo de imágenes.

Esto significa que se pueden crear potentes aplicaciones de tratamiento de imágenes, con código supervisado .Net, que usan menos memoria y funcionan más rápido al usar LEADTOOLS. Por otro lado, la Universidad de Las Ciencias Informáticas, ha adquirido una licencia de este producto para desarrollos investigativos, la cual respalda la decisión de utilizarla en la aplicación.

En este capítulo se llevó a cabo la elección de las herramientas y la metodología a utilizar desde la etapa de investigación y familiarización del problema, hasta el mismo proceso de desarrollo del software. Se fundamentó además, la utilización de los lenguajes C# 2.0 y UML 2.1; y se realizó una investigación del estado del arte a nivel mundial, en cuanto al procesamiento de imágenes y específicamente, a la detección de texto. Se mostró también la utilización de la detección de texto basado en análisis de textura y en bordes. Se justificó de forma precisa la elección del módulo OCR a utilizar.

CAPÍTULO 2: MÉTODO DE DETECCIÓN Y EXTRACCIÓN DE TEXTO PROPUESTO.

En este capítulo se analiza el método de detección y extracción de texto propuesto en esta investigación. Se describen las técnicas y algoritmos de procesamiento de imágenes digitales que conforman dicha metodología. Se realiza además una evaluación del método mediante la obtención y análisis de resultados experimentales.

2.1 Método de detección de texto propuesto.

El método de detección de texto propuesto está basado en bordes verticales. Estos métodos son robustos en la detección de textos de diferente color y tamaño. Además es significativo el bajo costo computacional y temporal que supone su implementación.

Este proceso consta de cuatro fases. Primeramente se detectan las posibles regiones que contienen texto artificial. Luego se generan las cajas de texto candidatas a contener los captions. Posteriormente se realiza la eliminación de falsos positivos mediante la aplicación de restricciones geométricas.

2.1.1 Detección de posibles regiones de texto.

El fotograma es llevado del espacio de colores RGB a escala de grises aplicando una sencilla transformación matemática. Se detectan los bordes verticales mediante el operador de Sobel (Molina 1998). Este operador emplea dos máscaras de convolución de tamaño 3x3. Estas máscaras están definidas por:

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & 1 \end{bmatrix} \qquad G_y = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & 1 \end{bmatrix}$$

Donde $\boldsymbol{G_x}$ es el kérnel para cambios horizontales y $\boldsymbol{G_y}$ para los verticales.

Como resultado de este proceso se obtiene una imagen binaria con los bordes verticales resaltados en blanco. El umbral para la binarización fue calculado mediante el método de Otsu (S. Morse 2002). Posteriormente se aplica una dilatación por un elemento estructurante (Molina 1998) de tamaño 1x11 quedando representadas en blanco las posibles regiones de texto.

Figura 6 a) Fotograma en escala de grises b) Mapa de bordes verticales c) Mapa de bordes dilatado.

2.1.2 Generación de cajas de texto candidatas.

Al mapa de bordes dilatados se le realiza un análisis de componentes conexas para agrupar las regiones de posible texto en palabras o líneas de texto. Para este proceso se utiliza el algoritmo de etiquetado de componentes conexas Run Length Code (Hur 2005).

Se calcula la proyección horizontal de cada región marcada como texto. Fijando un umbral en la media de la proyección horizontal se dividen las regiones que contienen más de una línea de texto, obteniéndose como resultado las regiones de texto candidatas.

Figura 7 a) Análisis de componentes conexas b) Proyección horizontal c) Regiones candidatas.

2.1.3 Filtrado de cajas de texto.

Cada región de texto candidata es analizada en cuanto a sus características geométricas. Se toman en cuenta el alto y la razón ancho entre alto. Una caja de texto es descartada si cumple con al menos una de las siguientes restricciones (Anthimopoulos, Gatos and Pratikakis 2006).

$$H_t < t_1 \qquad H_t > t_2 \qquad H_t/W_t > t_3$$

Donde H_t es el alto y W_t el ancho de la región. Siendo t_1 , t_2 y t_3 umbrales con valores de 10, alto del fotograma entre dos y 2 respectivamente.

Las cajas de texto que no son descartadas por restricciones geométricas son sometidas a un análisis de proyecciones verticales (Acharya and Ray 2005). En líneas de texto o palabras, la proyección vertical presenta numerosos picos espaciados uniformemente. Son desechadas las regiones que no cumplan con esta característica. Al final de este proceso se obtienen las cajas de texto finales.

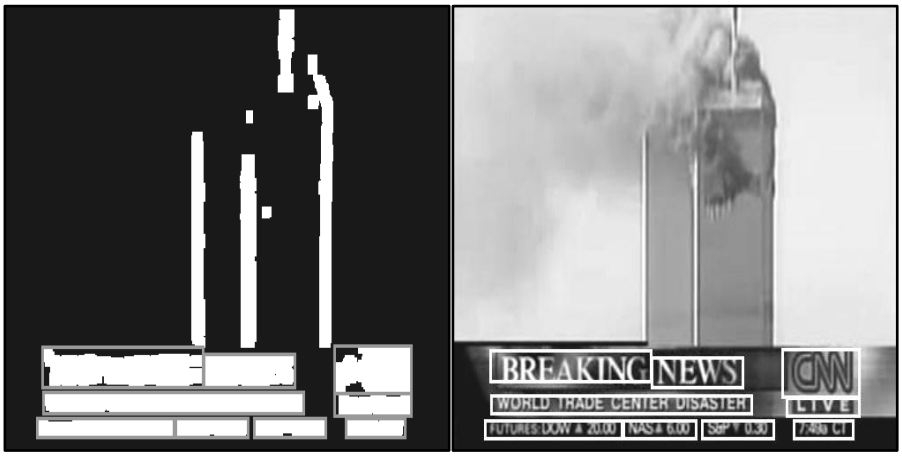


Figura 8 a) Regiones de texto luego del filtrado b) Cajas de texto en el fotograma.

2.2 Método de extracción de texto propuesto.

El texto artificial en los fotogramas de videos digitales puede encontrarse incrustado en fondos sumamente complejos. Conociendo además que los sistemas OCR (Sazaklis 2002) actuales solo manipulan eficientemente imágenes con texto en fondo limpio, se hace necesario binarizar cada una de las cajas de texto que se obtienen como resultado del proceso de detección de texto.

La binarización es realizada por el método de Otsu, el cual es un método de binarización adaptativa. Propone la división de la imagen en dos grupos o clases, el fondo de la imagen y los objetos que se encuentran en esta. Luego para cada umbral candidato se calcula la varianza entre las dos clases por la fórmula

$$\sigma(T) = n_f(T)n_o(T)\left[\mu_f(T) - \mu_o(T)\right]^2$$

Donde $\sigma(T)$ es la varianza entre las clases, $\mu_f(T)$ la media del fondo, $\mu_o(T)$ la media de los objetos, $n_f(T)$ y $n_o(T)$ el número de pixeles de fondo y de objetos respectivamente. El umbral óptimo T es el que maximiza la varianza entre las clases (S. Morse 2002).

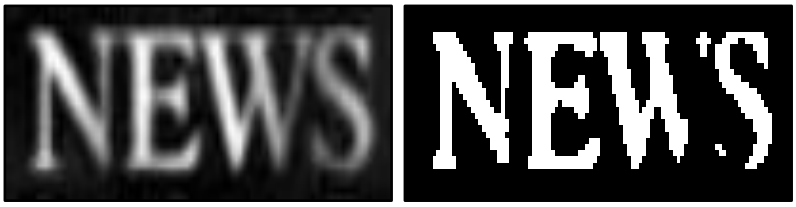


Figura 9 a) Caja de texto b) Caja de texto binarizada.

2.3 Resultados experimentales.

El índice de precisión y detección son dos criterios comúnmente usados para evaluar la eficiencia en los sistemas de recuperación de información (Yingzi and Chein-I 2003). Desde el punto de vista de la detección de texto, el índice de precisión cuantifica el porcentaje de cajas de texto correctas en el total de cajas de texto detectadas. Siendo el índice de detección el porciento de cajas de texto correctas en el total de cajas de texto presentes en los fotogramas. Formalmente están definidos por las siguientes ecuaciones:

$$\text{índice de precisión} = \frac{Tc}{Tc+Fp} \qquad \text{índice de detección} = \frac{Tc}{Tt}$$

Donde *Tc* es el número de cajas de texto correctas, *Fp es* el total de falsos positivos y *Tt* el total de cajas de texto.

Para evaluar experimentalmente el método de detección y extracción de texto propuesto se analizaron 120 fotogramas. Los fotogramas fueron extraídos de noticieros, películas, videos de deportes y

conferencias, garantizando la heterogeneidad de la muestra. En la Tabla 2.1 se muestran detalladamente los resultados experimentales obtenidos.

Parámetro	Valor
Fotogramas	120
Correctas	625
Falsos Positivos	142
Falsos Negativos	32
índice de precisión	81.3%
índice de detección	95.1%

Tabla 1 Resultados experimentales.

Se logra detectar el 95.1% del texto presente en los fotogramas. Nótese que en el método propuesto por Anthimopoulos, Gatos y Pratikakis (Anthimopoulos, Gatos and Pratikakis 2006) se obtiene solamente un 91.3% y en la mayoría de los artículos consultados este valor oscila entre 90 y 97%. Es preciso destacar además que la aparición de falsos positivos fue aumentado al diseñar el método de detección con el objetivo de lograr un mayor porcentaje de detección de cajas de texto.

En este capítulo se explica de forma detallada los métodos propuestos de detección y extracción del texto. Además son avalados por resultados experimentales que evidencian la efectividad de estos procedimientos e incluso los sitúan en un lugar aceptable, comparándolo con otros procedimientos similares. Se justifican las restricciones geométricas aplicadas a las cajas de texto candidatas; así como la utilización del algoritmo de etiquetado de componentes conexas Run Length Code y el método de Otsu para realizar la binarización.

CAPÍTULO 3: CARACTERÍSTICAS DEL SISTEMA.

3.1 Introducción

El desarrollo del Sistema para la Detección y Extracción de Texto en Videos Digitales (DETEXT) impone un gran reto tecnológico; pues, como es sabido, los procesos a automatizar se enmarcan en un campo de investigación relativamente joven. Por otro lado, el constante flujo y transformación de datos mediante filtros, que conlleva el proceso de tomar un video y extraer texto embebido, la necesaria reutilización de filtros y la ejecución concurrente de estos, son aspectos que se tienen en cuenta a la hora de desarrollar el sistema para lograr que sea eficiente .

Este capítulo define el objeto de automatización, así como se muestran los elementos que intervienen en la solución del problema y la relación entre ellos. Además se definen los actores y los casos de uso, se justifican los actores como futuros usuarios y se da una descripción detallada de los casos de uso. También se hará una descripción general de la propuesta del sistema que propone este trabajo de diploma y se hará referencia a algunos de los sistemas similares ya existentes. Se exponen además el modelo de negocio, la especificación de requerimientos funcionales y no funcionales del sistema y finalmente se mostrarán los diagramas de casos de uso.

3.2 Objeto de automatización

Los siguientes son procesos del negocio que serán objeto de automatización por el sistema:

3.2.1 Inserción de Videos

Proceso en el que se ingresan los videos digitales, situados en el repositorio, que se van a procesar, y se les extrae la información adjunta, como por ejemplo: nombre del video, formato, cantidad de fotogramas, frecuencia de imágenes por segundo, tamaño de las imágenes del video, entre otras.

3.2.2 Eliminación de Videos

Mediante este proceso se elimina el video que no se desea procesar, y se detiene el proceso de extracción del contenido textual en caso de estar siendo procesado.

3.2.3 Descomposición de Videos

Este proceso es el que se lleva a cabo cuando la persona que va a realizar la extracción del texto comienza a observar el video e identifica cuales son las imágenes que poseen texto.

3.2.4 Detección de Texto

Después de que se definió en qué imagen existe texto, entonces se define en qué región de estos fotogramas se encuentran específicamente.

3.2.5 Extracción de Texto

Es el proceso que se lleva a cabo una vez definida la región, el sujeto visualiza esta e identifica las palabras o frases encontradas, convierte a formato de texto (ASCII).

3.3 Propuesta de sistema.

Luego de realizar un profundo análisis del objeto de estudio, se ha concebido implementar un sistema que sea capaz de extraer el texto situado en las imágenes contenidas en los videos digitales.

El sistema es una aplicación de escritorio, que debe ser de poco costo y eficiente. DETEXT debe bridar la posibilidad de gestionar varios videos a la vez y procesarlos, logrando extraer la información contenida de cada uno de ellos. Debe poder además mostrar de forma detallada como se va realizando el proceso.

Existen varios sistemas que dentro de sus servicios utilizan motores de indexación similares a la solución aquí propuesta (Corsi 2007). Entre los más significativos se encuentran:

Google Video

Esta es la propia solución de Google (todavía Beta) para indexar y brindar acceso a la búsqueda de contenido de vídeo. Google indexa un número limitado de canales de transmisión comerciales y también vídeo clips. Google Video va a buscar las tomas que se aproximen más y las descripciones de texto de todos los videos en el archivo para resultados relevantes. Haga clic en un título del vídeo en su página de resultados y verá la imagen estática del vídeo y, cuando haya disponible una trascripción, recortes del texto trascrito. Disponible en http://video.google.com/.

ShadowTV

ShadowTV Webcasting permite a los emisores instantáneamente convertir material del video archivado y en vivo en contenido digital de video para su sitio web que permite ser buscado. Los emisores pueden explotar su propio contenido de video o sindicarlo para otros sitios. Consultar en http://www.shadowtv.com/.

BlinkxTV

A diferencia de otros proveedores de búsqueda, blinkx.tv no solamente le permite buscar utilizando palabras clave y consulta Booleana sino que también puede utilizar búsqueda conceptual. Este tipo de búsqueda solamente es brindada por blinkx.tv y le permite ingresar texto normal para el cual blinkx.tv le va a devolver resultados cuyo contenido sea conceptualmente similar a su texto de búsqueda.

Los requisitos funcionales se han capturado partiendo de la necesidad real existente en el ámbito de los procesos de extracción de información en videos, de manera que a la hora de implementar la solución, se evite caer en anti-patrones de software, lo que traerá consigo el rechazo y la no utilización de la solución informática. Disponible en http://www.blinkx.tv.

3.4 Modelo de negocio

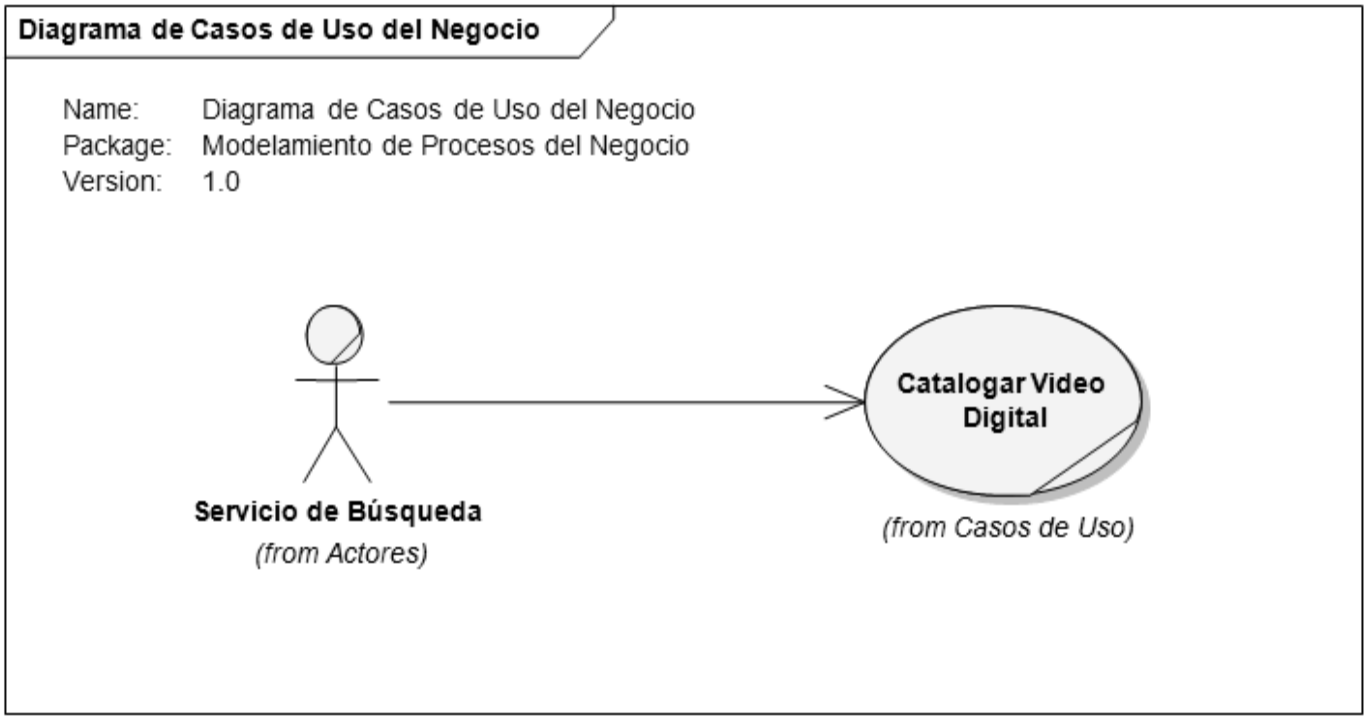
Los actores del negocio son las personas o sistemas que se benefician directamente con los procesos del negocio y obtienen un resultado de valor de los mismos. Se muestran a continuación:

Actores del negocio	Justificación
	Interviene en los procesos del negocio. Se beneficia directamente de todo el resultado del proceso de negocio.

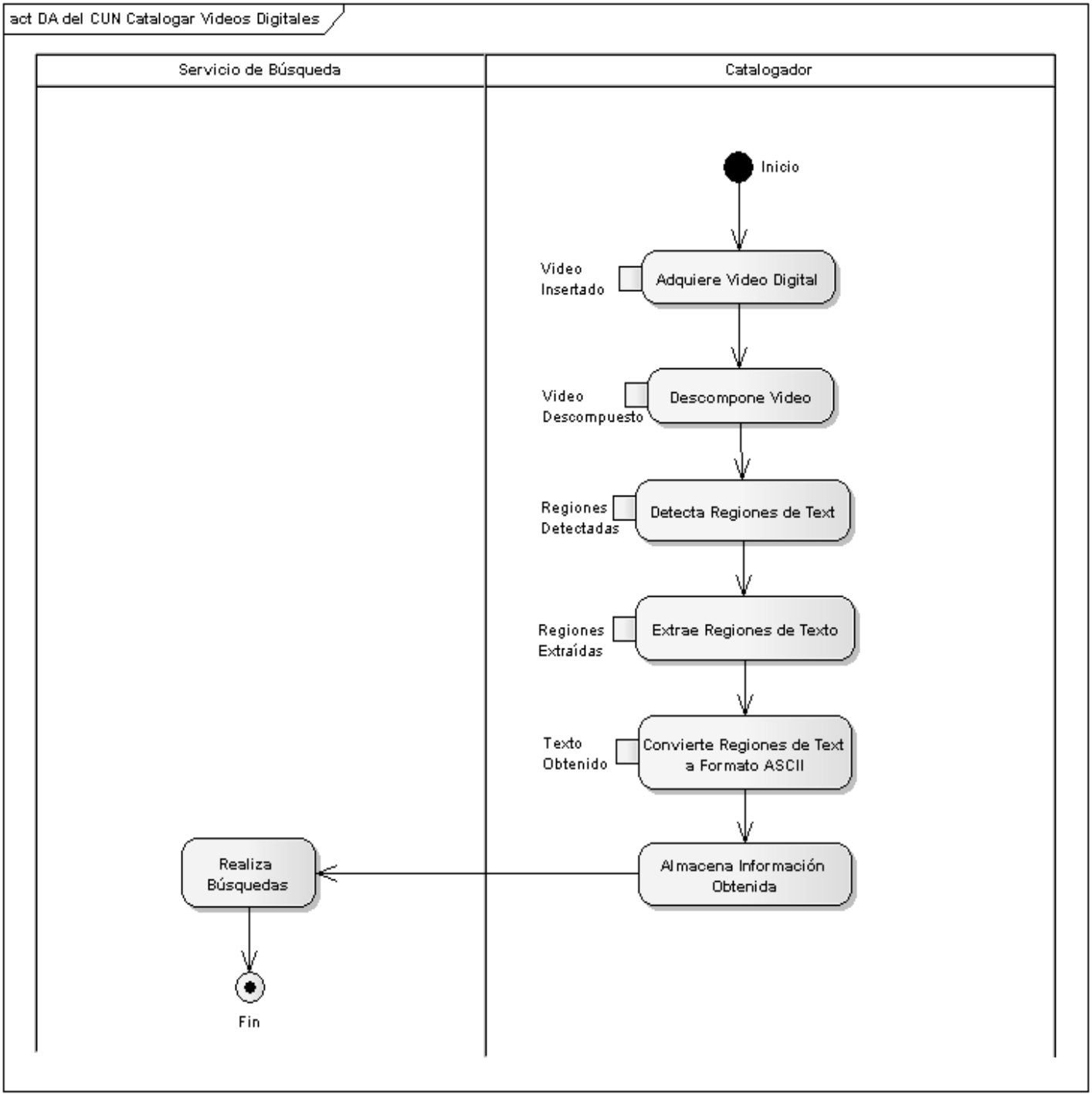
Los trabajadores del negocio son las personas o sistemas que rara vez reciben un beneficio, se encuentran involucrados en uno o más procesos del negocio, participan en ellos y son posibles candidatos a convertirse en actores del sistema.

Trabajadores del negocio	Justificación
	Interviene en todo el proceso de negocio en la gestión de los videos, en la detección y extracción del texto.

3.5 Diagrama de casos de uso del negocio

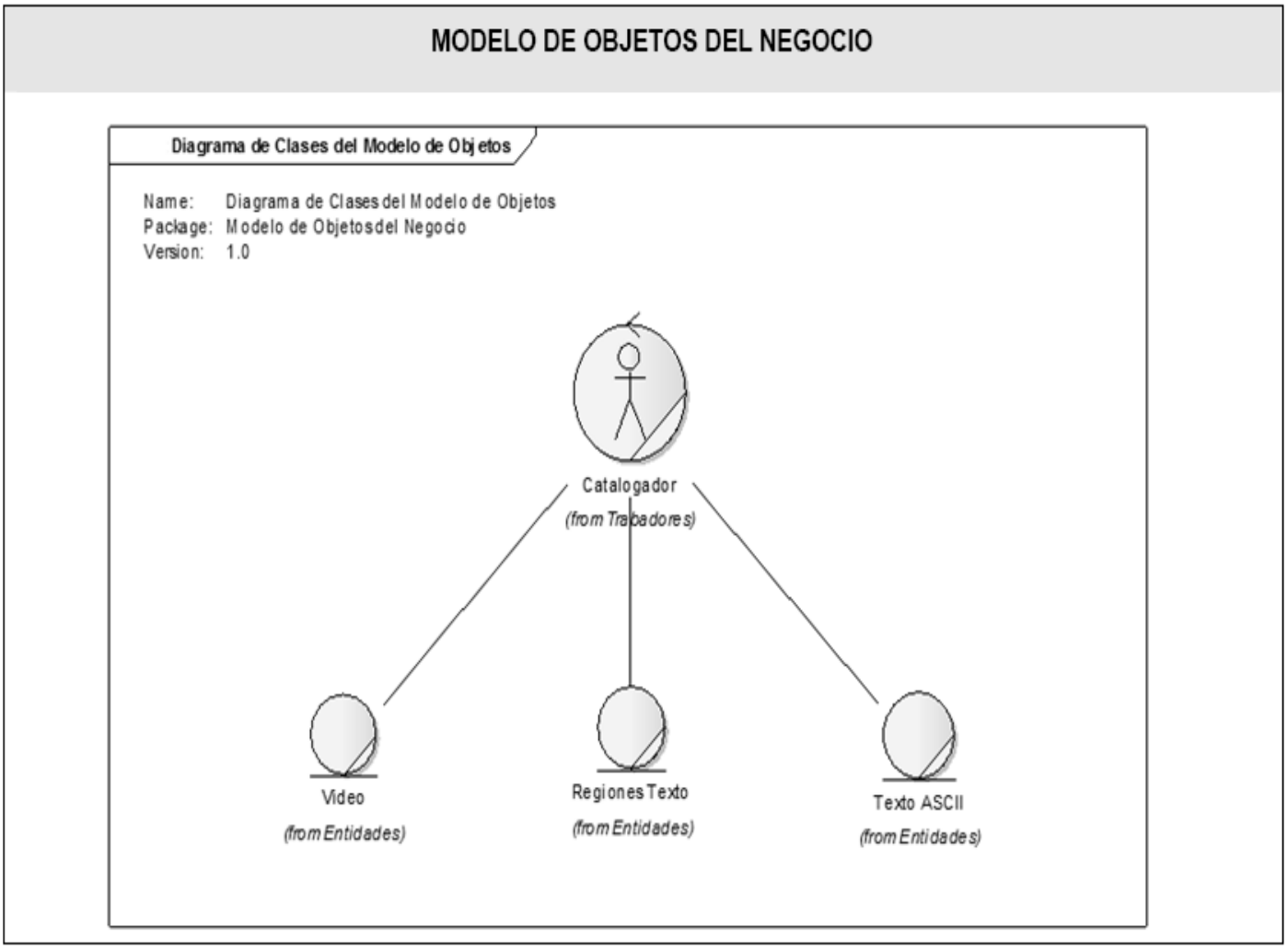


3.6 Diagramas de actividades de los Casos de Uso del negocio



3.7 Diagrama de Clases del Modelo de Objetos

El modelo de objetos del negocio expresa las relaciones que existen entre los trabajadores del negocio y las entidades del mismo.



3.8 Requerimientos Funcionales.

Los requerimientos funcionales del sistema expresan las operaciones que éste debe implementar para satisfacer a los clientes. El sistema aquí expuesto consta de tres requerimientos, uno de ellos representan paquetes de requerimientos: Gestionar Videos, que se divide en las operaciones de Insertar, Eliminar, Descomponer y Obtener Datos.

1. **Gestionar Videos**

   1.1 Insertar Video.

   1.2 Eliminar Videos.

   1.3 Obtener Datos Adjuntos del Video.

2. **Procesar Video.**

3. **Mostrar Detalles del Proceso**

3.9 Requerimientos no funcionales del sistema

1. Usabilidad.

   El sistema debe ser lo más atractivo posible y que facilite el trabajo a los catalogadores. La realización de sus operaciones debe ser fácil y rápida, los usuarios no deberán poseer grandes conocimientos de informática para poder operar con este.

2. Rendimiento.

   El sistema debe ser capaz de procesar múltiples videos a la vez, evitando demoras innecesarias en el servicio.

3. Políticos - culturales.

   El sistema actualmente solamente está disponible en el idioma español, pero en un futuro está pensado implementar las versiones correspondientes para otros idiomas. Los logotipos e imágenes usadas están en correspondencia con el tipo de institución donde será usado.

4. Portabilidad.

   El sistema debe ser capaz de ejecutarse sobre la plataforma libre.

5. Legales.

   El sistema y toda la documentación generada con el mismo, pertenecen a la Universidad de la Ciencias Informáticas (UCI).

6. Confiabilidad.

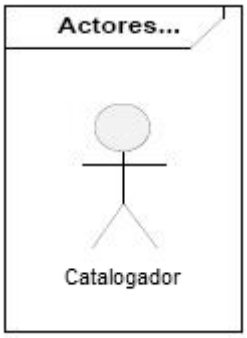
   El sistema debe estar bien documentado, de esta forma se garantiza que el tiempo de mantenimiento sea mínimo.

7. Interfaz.

   La aplicación propuesta poseerá una interfaz sencilla, amigable y cómoda para los usuarios a quien va dirigida. La interfaz deberá ser lo suficientemente intuitiva como para permitir que usuarios con conocimientos básicos puedan operar el sistema sin mayores contratiempos.

3.10 Definición de los casos de uso

Definición de los actores

Actores del Sistema	Justificación
	Es la persona con la cual interactúa el sistema. Es el único rol existente, inicia todos los casos de usos y posee el mayor privilegio.

Listado de casos de uso.

CU-1	Insertar Video
Actor	Catalogador
Descripción	Este caso de uso se inicia cuando el catalogador desea procesar un video digital, proveniente de un repositorio. Es entonces cuando lo introduce al sistema, dejándolo listo para su procesamiento.
Referencia	RF-1

CU-2	Eliminar Video
Actor	Catalogador
Descripción	Este caso de uso se inicia cuando el catalogador desea eliminar un video, que está en la lista de videos a procesar del sistema o se está procesando. El sistema detiene su procesamiento, en caso de estar procesando, y lo elimina de inmediato.
Referencia	RF-1

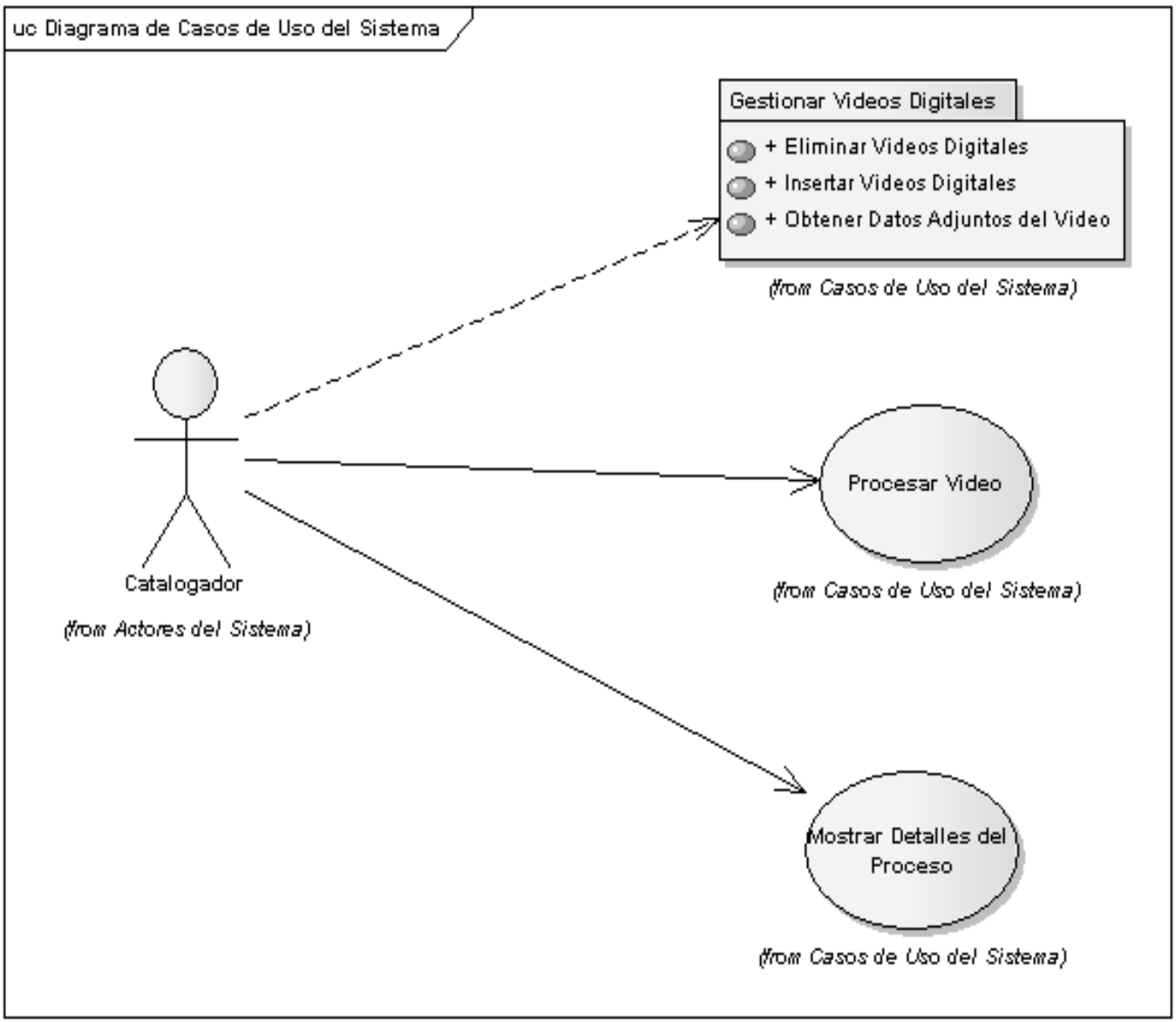
CU-3	Obtener datos Adjuntos del Video
Actor	Catalogador
Descripción	Este caso de uso se inicia cuando el catalogador solicita ver las características del video. De esta forma se extraen los datos que posee el video digital internamente, tales como: frecuencia de imágenes por segundo, tamaño, duración, formato, compresión, bitrate, entre otros.
Referencia	RF-1

CU-4	Procesar Video
Actor	Catalogador
Descripción	Este caso de uso se inicia cuando el catalogador decide procesar uno o varios videos a la vez, permitiendo extraer el texto en formato ASCII de cada uno de los videos procesados.
Referencia	RF-2

CU-5	Mostrar Detalles del Proceso
Actor	Catalogador
Descripción	Este caso de uso se inicia cuando se selecciona un video que está en procesamiento y se decide ver detalladamente lo que ocurre, mostrándose el fotogramas que se está gestionando en ese momento, las regiones detectadas, el texto extraído entre otras cosas.
Referencia	RF-3

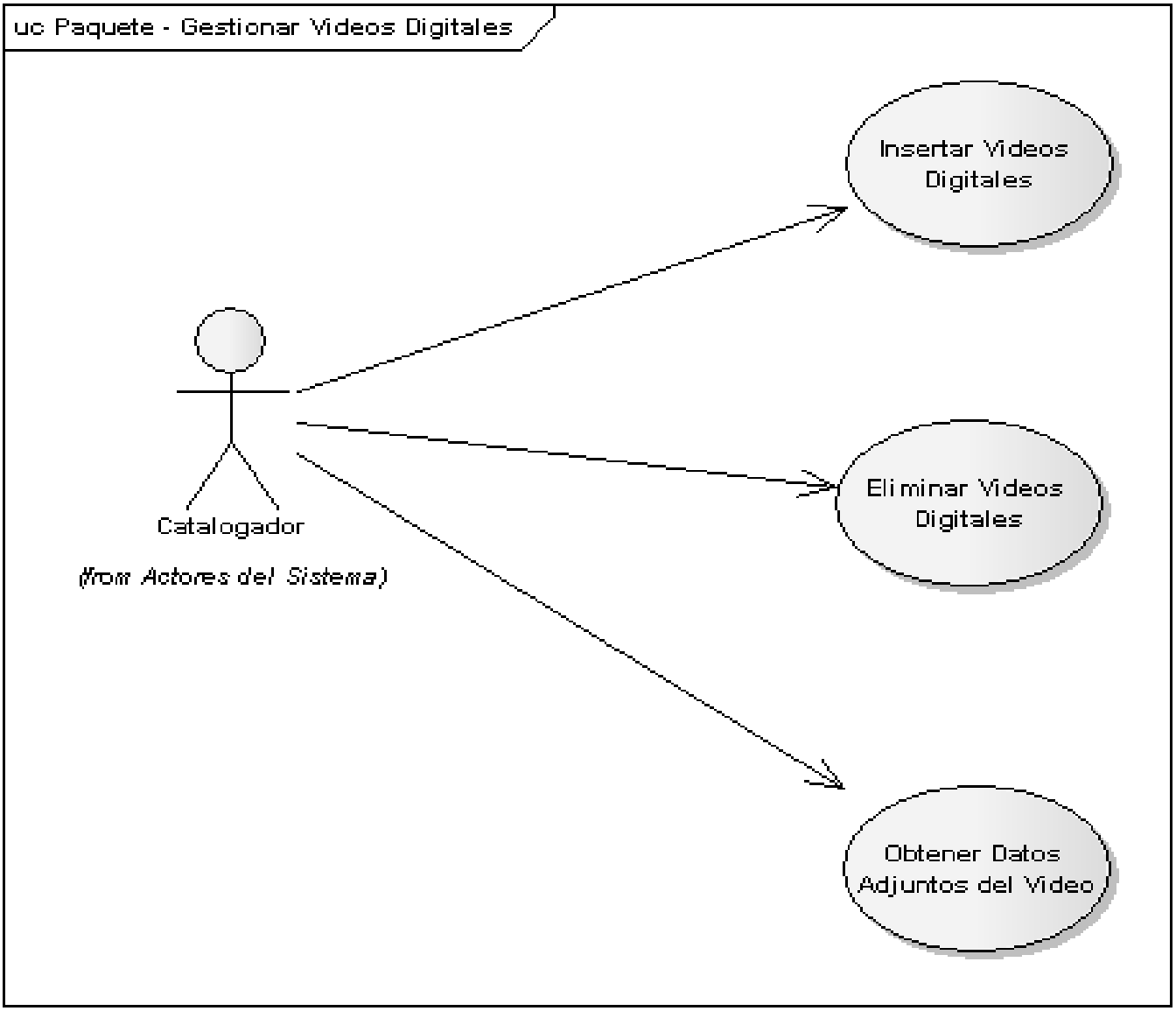
3.11 Diagrama de casos de uso del sistema

DIAGRAMA DE CASOS DE USO DEL SISTEMA



Por razones de claridad se ha creado el paquete Gestionar Videos Digitales. A continuación se muestra el diagrama de casos de uso de este paquete.

DIAGRAMA DE CASOS USO DEL PAQUETE GESTIONAR VIDEOS DIGITALES



Casos de uso por ciclos de desarrollo

Primer ciclo de desarrollo

Código	Nombre del caso de uso	Paquete	Justificación de la Selección
CU1	Insertar Video	Gestionar Videos	Es necesario para la realización del CU4.
CU3	Obtener Datos Adjuntos del Video	Gestionar Videos	Este caso de uso es fundamental, pues determina la información que se va a utilizar en el proceso posterior, pero además le brinda al usuario un nivel básico de información sobre el video.
CU4	Procesar Video	---	Es el caso de uso más importante pues sin él, el sistema no tendría sentido. Se encarga de extraer el texto del Video.

Segundo ciclo de desarrollo

Código	Nombre del caso de uso	Paquete	Justificación de la Selección
CU2	Eliminar Video	Gestionar Videos	Permite eliminar un video en cualquier momento.
CU5	Mostrar Detalles del Proceso	---	Es menos importante que los CU del primer ciclo, pero sin el sistema dejaría de brindar un servicio muy significativo e informativo.

3.12 Descripción extendida de los Casos de Uso.

Nombre del Caso de Uso:	Insertar Video
Actores:	Catalogador (Inicia)
Propósito:	Introducir al sistema el/los videos digitales a procesar.
Resumen:	El Catalogador toma un video y lo arrastra hacia la aplicación, donde se cargar el video, en caso de tener el formato correcto y no estar corrupto.
Referencias:	REQ:1.01
Tipo:	Crítico
Precondiciones:	El formato de el/los videos que se van a añadir deben ser *.avi o *.mpg.

Curso Normal de los eventos	
Acciones de los Actores	Respuesta del Sistema
1- Toma el video y lo añade a la aplicación	2- Acepta el archivo de video.
	3- Analiza el video y si tiene el formato correcto lo inserta en la lista a procesar.
	4- Muestra el video dentro de la lista de videos en proceso en el orden en que fueron añadidos.
5- Observa la lista.	
Curso Alterno de los eventos	
Línea 3: a) Si el video tiene un formato no compatible con los especificados o esta corrupto, muestra una notificación de que no se pudo abrir.	
Postcondiciones	El video esta listo para procesar.

Nombre del Caso de Uso:	Eliminar Video
Actores:	Catalogador (Inicia)
Propósito:	Eliminar de la lista del sistema el/los videos digitales a procesar, procesándose o procesados.
Resumen:	El Catalogador selecciona un video y lo elimina, el sistema detiene el procesamiento en caso que se este procesando y lo elimina de la lista de videos.
Referencias:	REQ:1.02
Tipo:	Primario
Precondiciones:	El video debe haber sido insertado.

Curso Normal de los eventos	
Acciones de los Actores	Respuesta del Sistema
1- Selecciona el video y lo elimina	2-Verifica si el video está procesándose.
	3- Si no está procesándose, lo elimina de la lista de videos a procesar.
	4- Muestra la lista de videos sin el video que se eliminó.
5- Observa la lista.	

Curso Alterno de los eventos	
Línea 2: a) Si el video está procesándose el sistema detiene el procesamiento y lo elimina de la lista de videos.	
Postcondiciones	El video ya no está en el la lista de videos a procesar.

Nombre del Caso de Uso:	Obtener Datos Adjuntos del Video
Actores:	Catalogador (Inicia)
Propósito:	Extraer información del video
Resumen:	El Catalogador selecciona un video y solicita ver su información; la aplicación toma la información y muestra los datos internos del video.
Referencias:	REQ:1.03
Tipo:	Crítico
Precondiciones:	El video debe haber sido insertado.

Curso Normal de los eventos	
Acciones de los Actores	Respuesta del Sistema
1- Selecciona el video y solicita sus datos.	2- Recibe petición.
	3- Accede a la información adjunta del video.
	4- Muestra la información obtenida.
5- Observa la información.	
Postcondiciones	Se muestra la información adjunta del video.

Nombre del Caso de Uso:	Procesar Video
Actores:	Catalogador (Inicia)
Propósito:	Procesar el video para extraer el texto contenido en sus fotogramas.
Resumen:	El Catalogador selecciona uno o más videos y los manda a procesar. La aplicación toma el video y lo procesa extrayendo el texto contenido en sus fotogramas.
Referencias:	REQ:2.0
Tipo:	Crítico
Precondiciones:	El video debe haber sido insertado.

Curso Normal de los eventos	
Acciones de los Actores	Respuesta del Sistema
1- Selecciona el video o los videos y los manda a procesar.	2- Recibe petición.
	3- Descompone el video en fotograma.
	4- Extrae los fotogramas significativos.
	5- Detectar regiones candidatas a texto en cada fotograma.
	6- Extrae regiones candidatas
	7- De las regiones extraídas saca el texto en formato ASCII.
5- Observa el texto extraído.	
Postcondiciones	El video ha sido catalogado.

Nombre del Caso de Uso:	Mostrar Detalles del Proceso
Actores:	Catalogador (Inicia)
Propósito:	Mostrar como se está realizando el proceso de detección y extracción de texto.
Resumen:	El Catalogador toma un video que se esté procesando y solicita ver su procesamiento, la aplicación muestra una serie de procesos que están ocurriendo en ese instante.
Referencias:	REQ:3.0
Tipo:	Primario
Precondiciones:	El video debe estarse procesando.

Curso Normal de los eventos	
Acciones de los Actores	Respuesta del Sistema
1- Selecciona el video y solicita ver detalles del proceso.	2- Verifica si el video está en procesamiento.
	3- Si el video está procesándose, muestra el histograma del fotograma actual en proceso, muestra además la imagen binarizada, detección de bordes verticales, las regiones detectadas, y el texto que se extrajo de dicha región.
5- Observa la Información mostrada.	
Curso Alterno de los eventos	
Línea 2: a) Si el video no está en procesamiento muestra una notificación informando que para mostrar el proceso el video este se debe estar procesando.	
Postcondiciones	Se muestran las ocurrencias del proceso.

En el presente capítulo se han analizado los procesos de negocio que serán objeto de automatización. Se describió el sistema propuesto, exponiendo el modelo del negocio del mismo, los requerimientos funcionales y no funcionales, mostrándose además los diagramas de casos de uso que se han modelado. Además, se realizó una descripción detallada de cada caso de uso.

CAPÍTULO 4: ANÁLISIS Y DISEÑO DEL SISTEMA

En el capítulo, se definen las clases de análisis y diseño del software de los casos de usos del primer ciclo de desarrollo; así como los diagramas de secuencias cumpliendo con las descripciones extendidas de los casos de uso. Se expone la arquitectura del sistema, favoreciendo el cumplimiento de los requisitos no funcionales. Se describen además las clases entidades y las controladoras del flujo de trabajo análisis y diseño.

4.1 Modelo arquitectónico.

La necesidad de establecer un marco de trabajo estructural básico, o sea, la estructura general del software, ofreciendo una integridad conceptual al sistema, facilitando la comunicación entre los diferentes participantes en el desarrollo y aportando una visión de estructura e interacción de los componentes del sistema; es lo que da vital importancia a la definición de un modelo arquitectónico.

DETEX, es un sistema con una arquitectura centrada en los flujos de datos, basada en el patrón “pipe and filter” (tuberías y filtros); donde los componentes presentes en el sistema responden a decisiones arquitectónicas para hacer cumplir requerimientos funcionales y no funcionales. Estos componentes están asociados a flujos de datos y refinamientos sucesivos. O sea, un conjunto de elementos denominados “filtros” conectados entre si por “tuberías” trasmiten datos desde un componente al siguiente. Cada filtro trabaja de manera independiente de los componentes. Se diseñan de tal modo que esperan un conjunto de datos en un determinado formato y obtiene como resultado otros datos de salida en un formato específico.

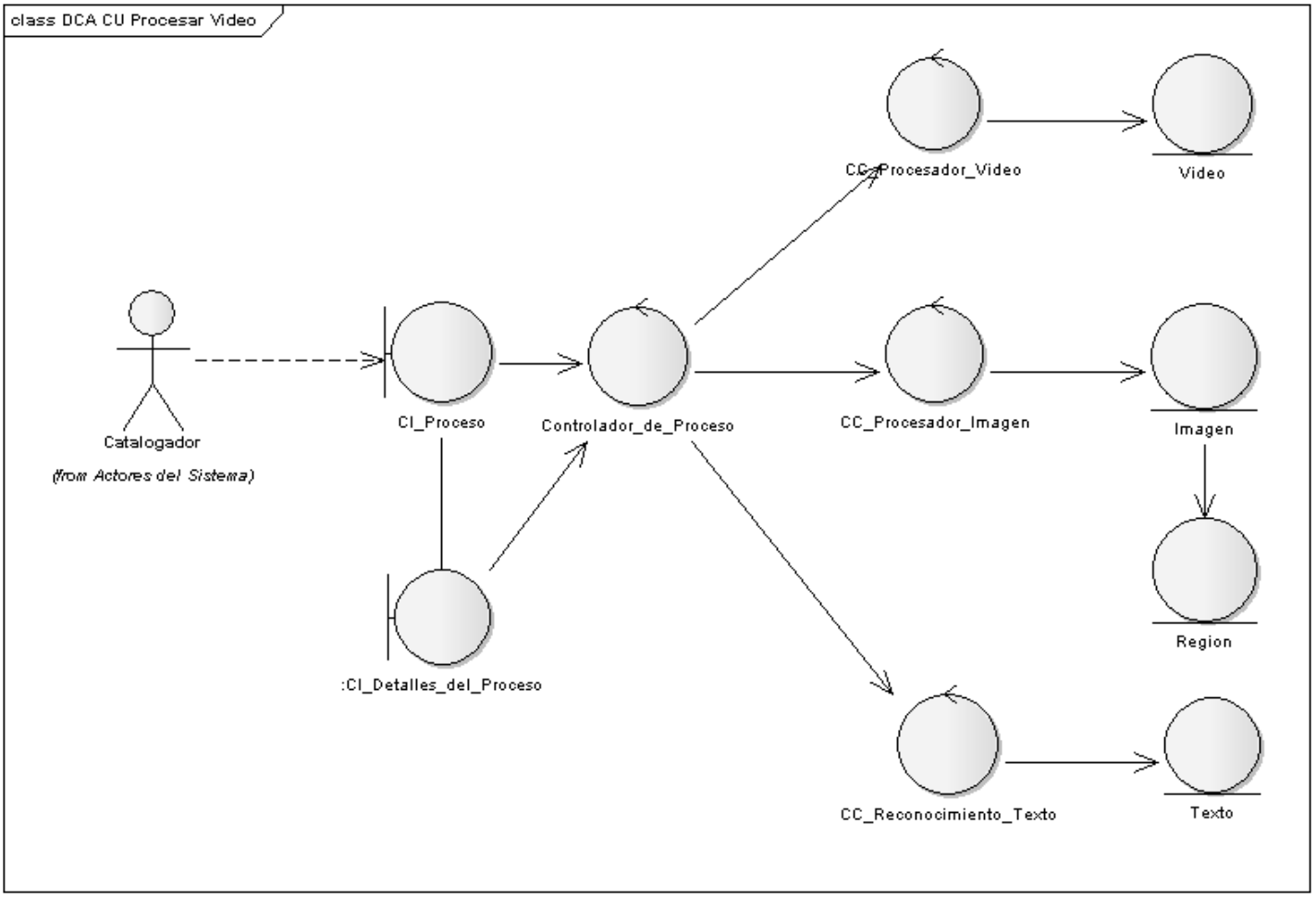
Este patrón de arquitectura es particularmente efectivo a la hora de descomponer el problema en pasos independientes, reutilizar de filtros, facilitar el mantenimiento, independencia entre filtros y ejecución concurrente de filtros.

4.2 Modelo Análisis.

Con el propósito de conseguir una comprensión más precisa y una descripción más detallada del sistema, se refinan y estructuran los requisitos obtenidos con anterioridad. Se han realizado los diagramas de clases de análisis, los cuales muestran qué clases participan en las realizaciones de los distintos casos de usos del primer ciclo de desarrollo planteados en capítulo anterior. Estos diagramas se muestran a continuación.

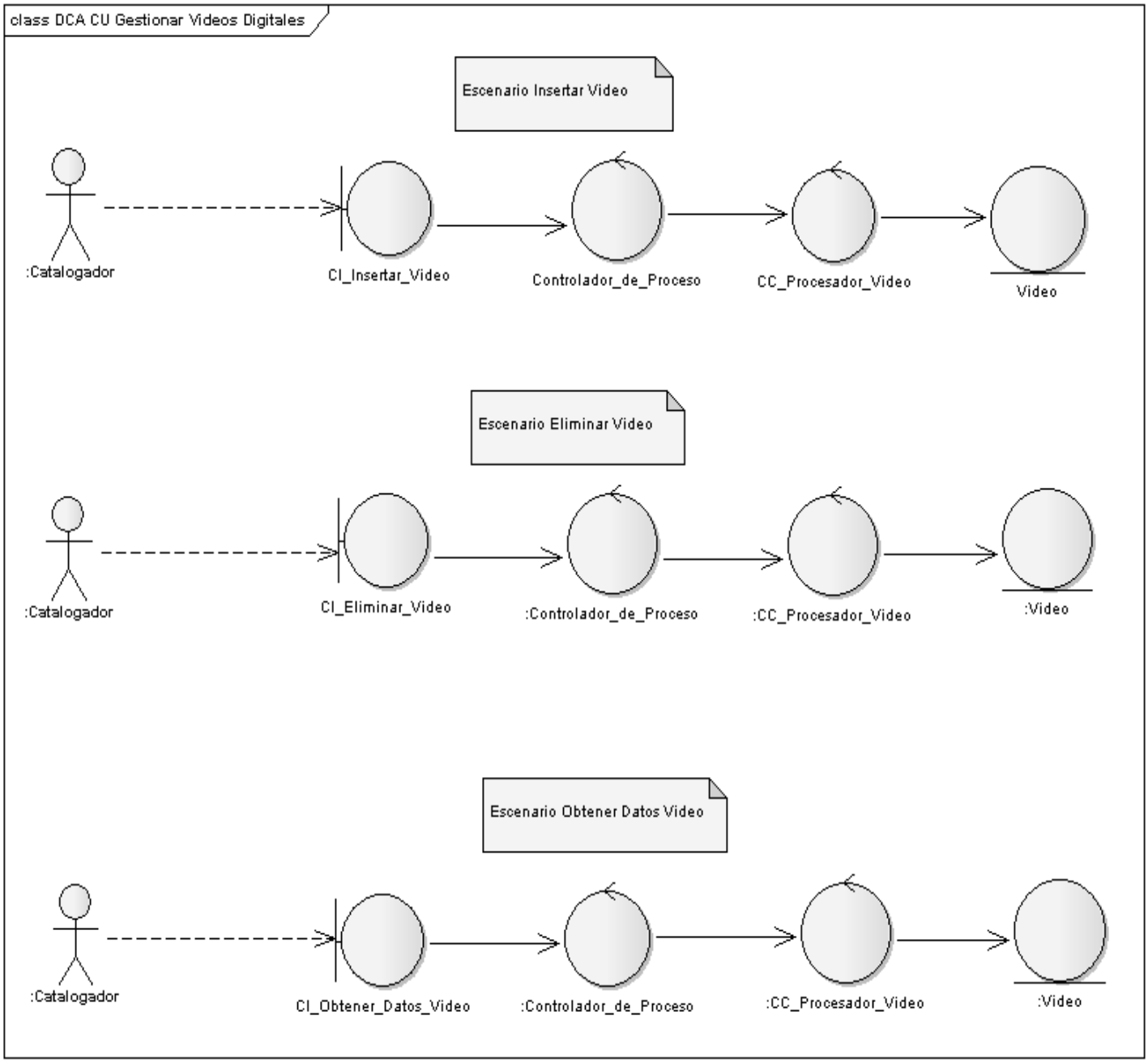
4.2.1 Diagrama de clases Procesar Videos Digitales

DIAGRAMA DE CLASES DE ANÁLISIS CU PROCESAR VIDEOS DIGITALES



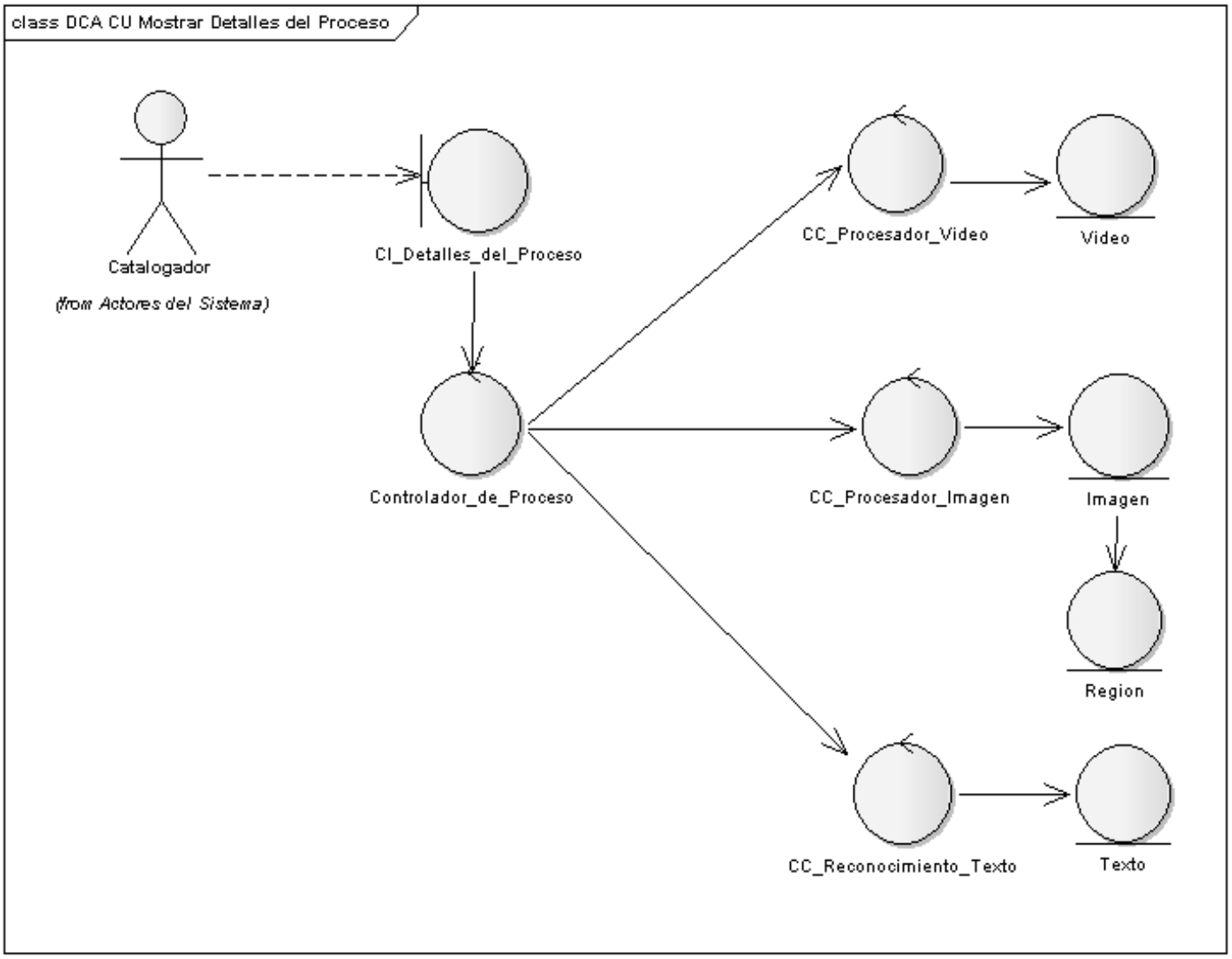
4.2.2 Diagrama de clases Gestionar Videos Digitales

DIAGRAMA DE CLASES DE ANÁLISIS CU GESTIONAR VIDEOS DIGITALES



4.2.3 Diagrama de clases Mostrar Detalles del Proceso

DIAGRAMA DE CLASES DE ANÁLISIS CU MOSTRAR DETALLES DEL PROCESO



4.3 Modelo Diseño.

Para la creación de una arquitectura estable y sólida, y de un plano del modelo de implementación; durante esta fase se analiza a profundidad si es posible dar una solución que satisfaga a los requerimientos significativos.

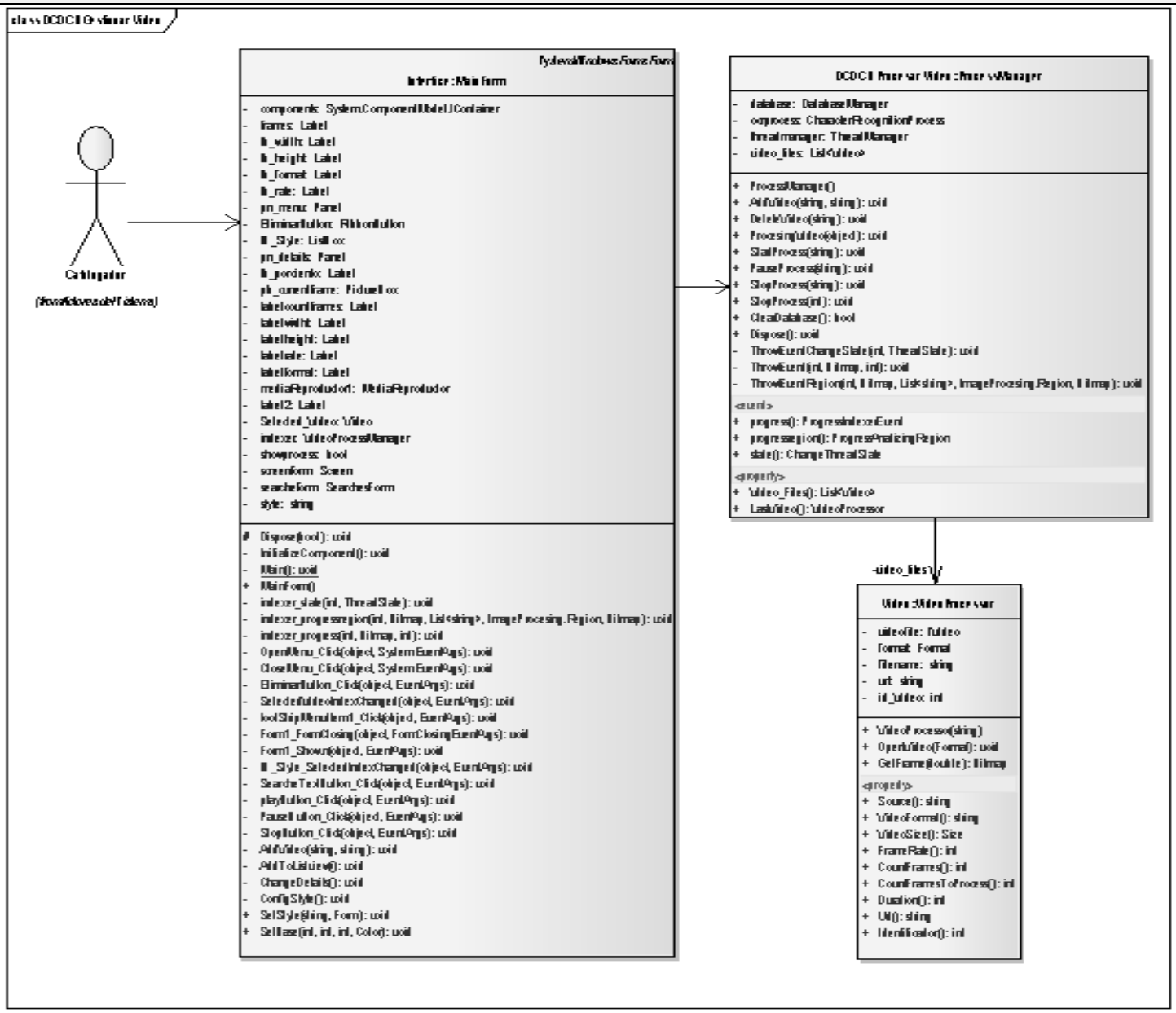
En el diseño se modela el sistema y se encuentra su forma para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Básicamente es adaptar el diseño para que se corresponda con el entorno de implementación, diseñando sus funcionalidades.

En esta etapa se generan como principales artefactos los diagramas de clases de diseño, así como los diagramas de interacción, ya sean de secuencia o de colaboración; los cuales se utilizan para modelar los aspectos dinámicos de un sistema utilizando un conjunto de objetos y sus relaciones e incluyendo los mensajes que se pueden enviar entre ellos.

Diagrama de clases del diseño

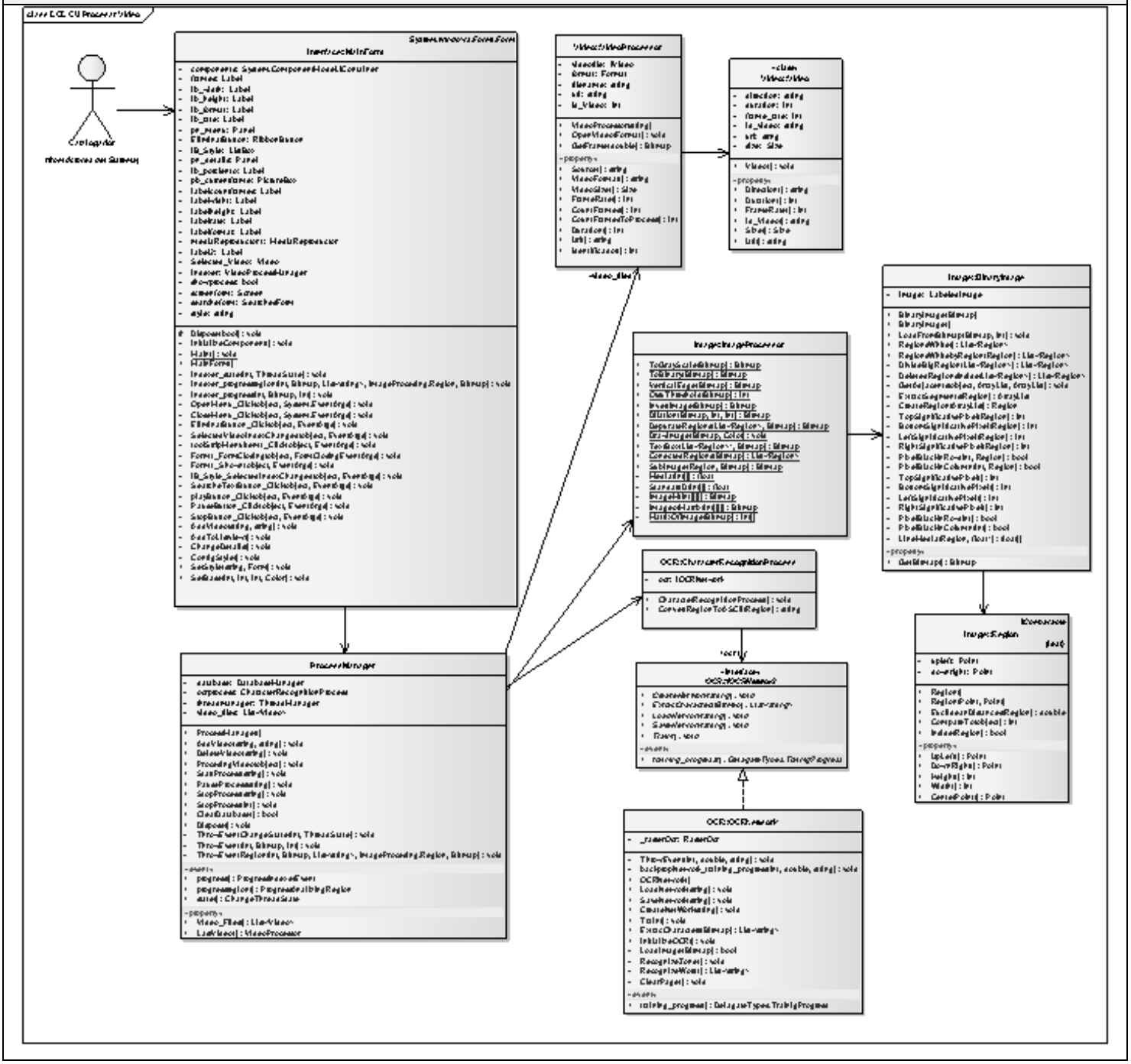
4.3.1 Diagrama de clases del diseño Gestionar Videos Digitales

DIAGRAMA DE CLASES DE DISEÑO CU GESTIONAR VIDEOS DIGITALES



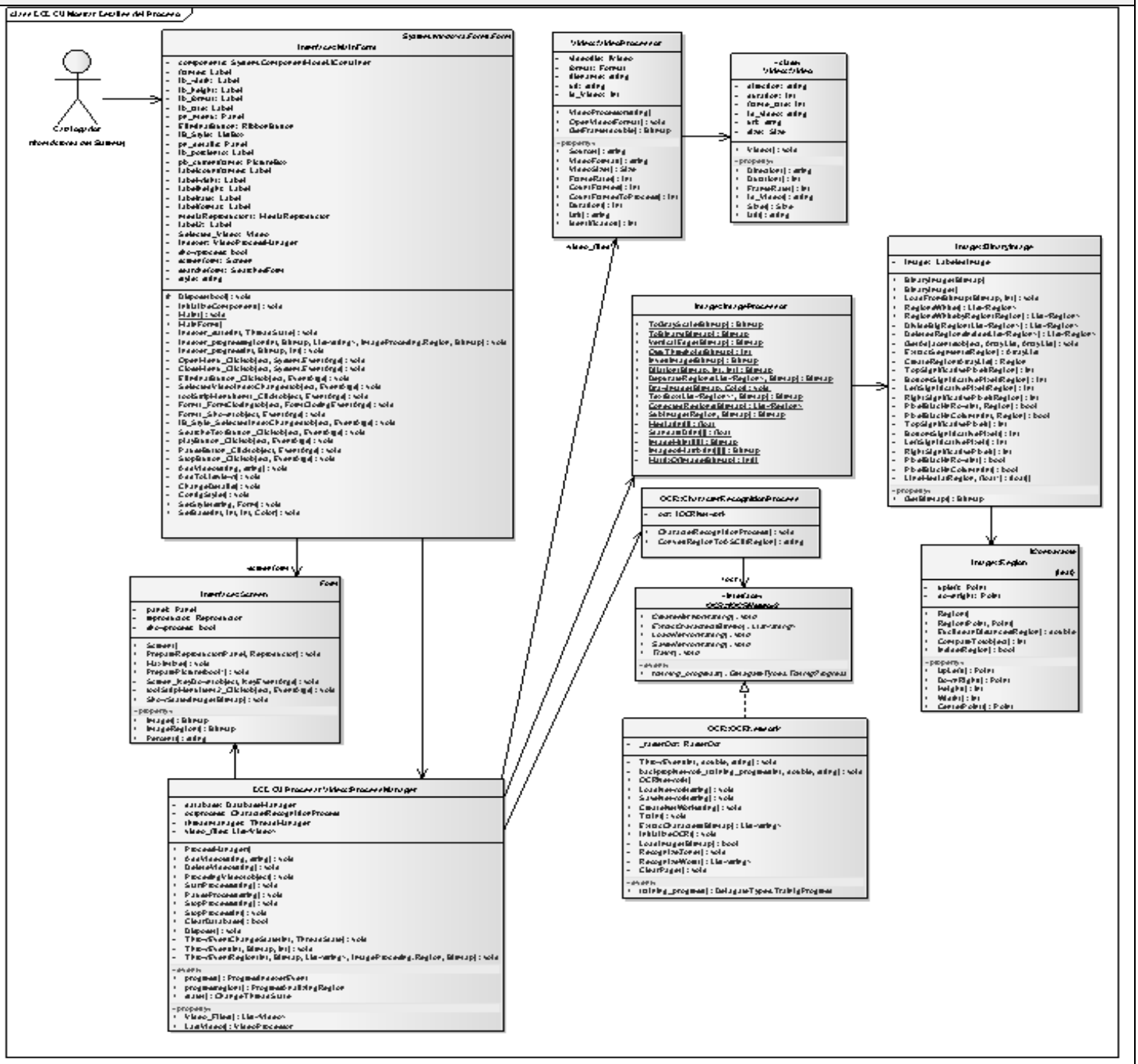
4.3.2 Diagrama de clases del diseño Procesar Videos Digitales

DIAGRAMA DE CLASES DE DISEÑO CU PROCESAR VIDEOS DIGITALES



4.3.3 Diagrama de clases del diseño Mostrar Detalles del Proceso

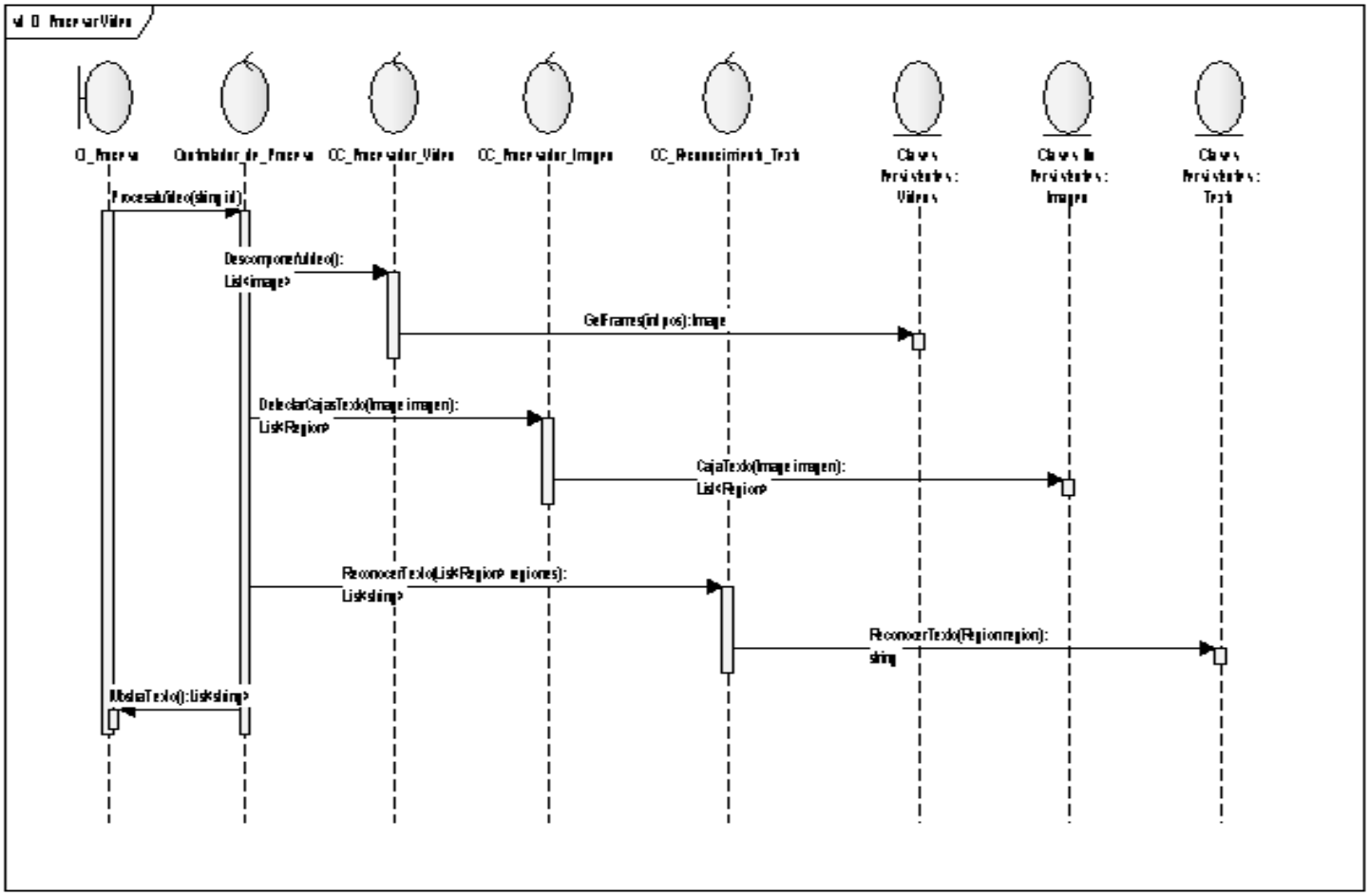
DIAGRAMA DE CLASES DE DISEÑO CU MOSTRAR DETALLES DEL PROCESO



Diagramas de Secuencia

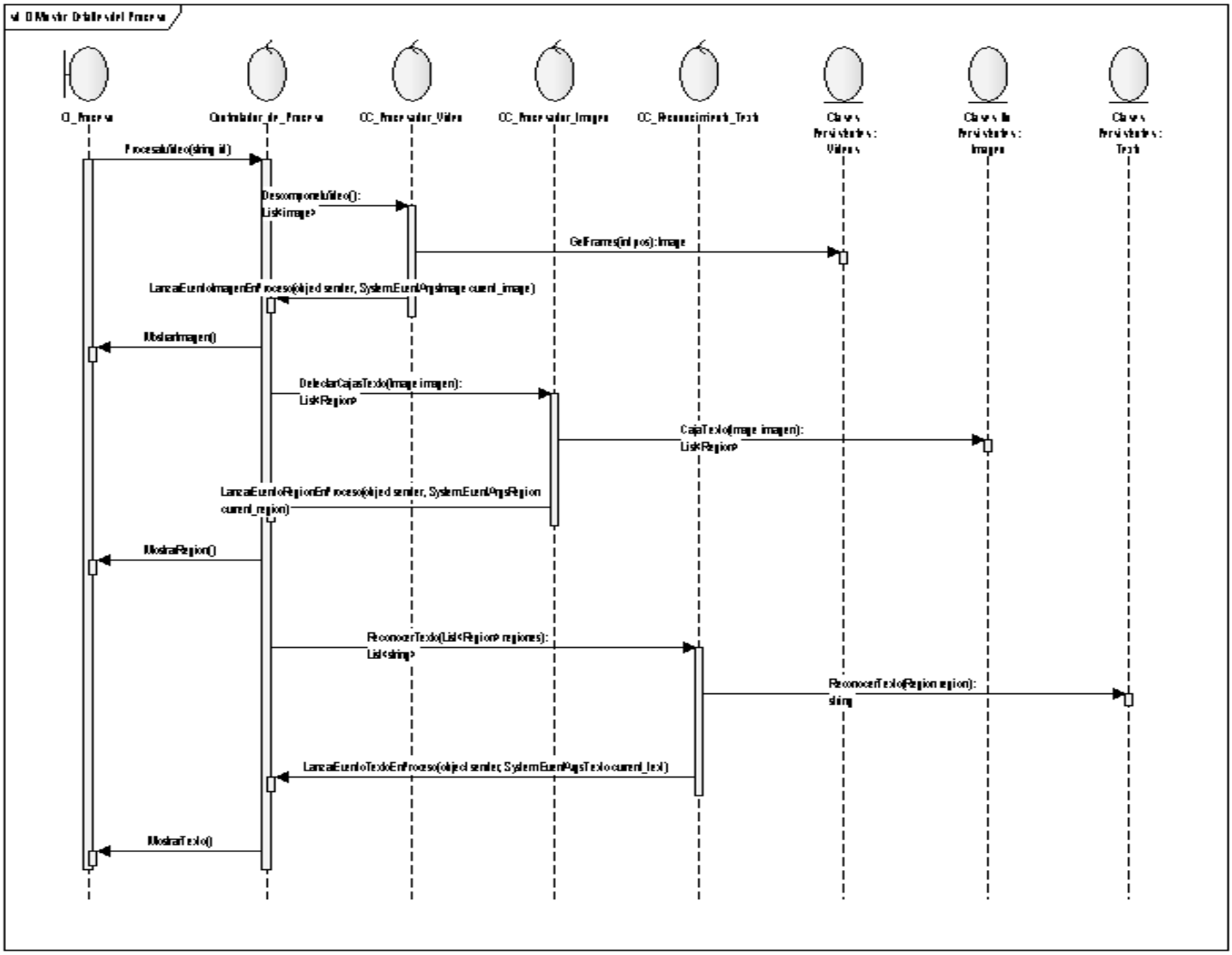
4.3.4 Diagrama de Secuencia Procesar Videos Digitales

DIAGRAMA DE SECUENCIA CU PROCESAR VIDEOS DIGITALES



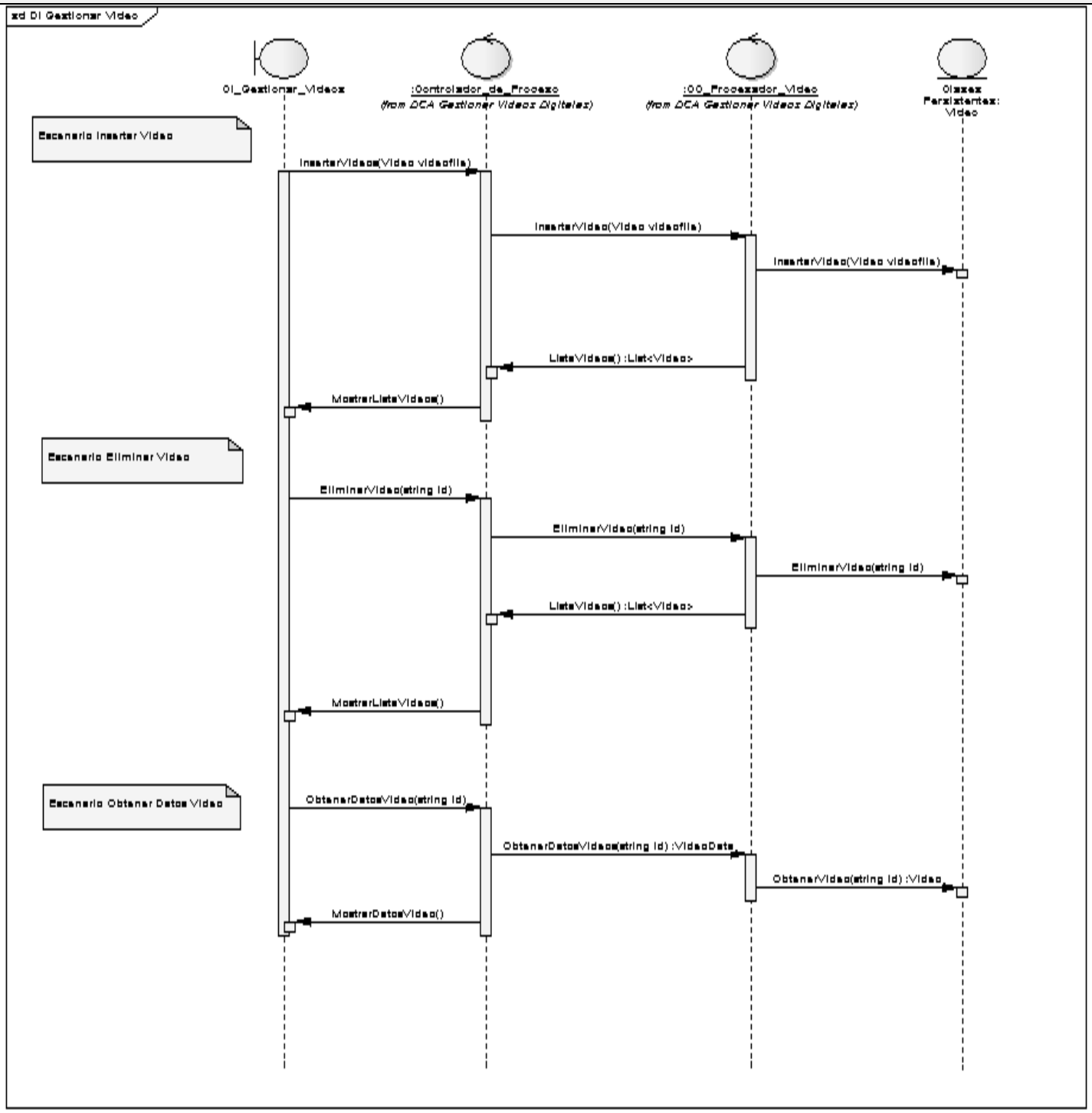
4.3.5 Diagrama de Secuencia Mostrar Detalles del Proceso

DIAGRAMA DE SECUENCIA CU MOSTRAR DETALLES DEL PROCESO



4.3.6 Diagrama de Secuencia Gestionar Videos Digitales

DIAGRAMA DE SECUENCIA CU GESTIONAR VIDEOS DIGITALES



4.4 Descripción de las clases.

4.4.1 Clase Video

Nombre	Video	
Tipo clase	Entidad	
Atributo		Tipo
direction		string
duration		int
frame_rate		int
url		string
id_video		int
size		Size
Para cada responsabilidad		
Nombre	Video()	
Descripción	Constructor	
Nombre	Direction()	
Descripción	Obtener la dirección física donde se encuentra el video.	
Nombre	Duration()	
Descripción	Obtener la duración del video en segundos.	
Nombre	FrameRate()	
Descripción	Obtener la frecuencia de imágenes por segundo transmitida por el video.	
Nombre	Id_Video()	
Descripción	Obtener el id del video.	
Nombre	Size()	
Descripción	Obtener el ancho y largo de las imágenes del video.	
Nombre	Url	
Descripción	Obtener la dirección url donde se encuentra publicado el video.	

4.4.2 Clase BinaryImage

Nombre	BinaryImage
Tipo clase	Entidad
Atributo	Tipo
Image	LabeledImage
Para cada responsabilidad	
Nombre	BinaryImage (ImageBinarized: Bitmap)
Descripción	Constructor
Nombre	BinaryImage ()
Descripción	Constructor
Nombre	LoadFromBitmap (bitmap: Bitmap, threshold: int)
Descripción	Convierte, usando un umbral, un bitmap en una BinaryImage.
Nombre	RegionsWhite()
Descripción	Obtener todas las regiones conexas.
Nombre	GetBitmap ()
Descripción	Obtener un Bitmap a partir de una BinaryImage.

4.4.3 Clase Region

Nombre	Region	
Tipo clase	Entidad	
Atributo		Tipo
upleft		Point
downright		Point
Para cada responsabilidad		
Nombre	Region	
Descripción	Constructor	
Nombre	Region(upleft: Point, downright: Point)	
Descripción	Constructor	
Nombre	UpLeft ()	
Descripción	Obtiene el punto superior izquierdo de la región.	
Nombre	DownRight ()	
Descripción	Obtiene el punto inferior derecho de la región.	
Nombre	Height ()	
Descripción	Alto de la región.	
Nombre	Width ()	
Descripción	Ancho de la región.	
Nombre	CenterPoint ()	
Descripción	Obtiene el punto centro de la región.	
Nombre	EuclideanDistances (region: Region)	
Descripción	Calcula la distancia Euclideana existente entre los puntos centros de dos regiones.	
Nombre	CompareTo ()	
Descripción	Compara si dos regiones son iguales.	

4.4.4 Clase VideoProcessor

Nombre	**VideoProcessor**
Tipo clase	Controladora
Atributo	Tipo
videofile	IVideo
format	Format
filename	string
url	string
id_video	int
Para cada responsabilidad	
Nombre	VideoProcessor(file_name: string)
Descripción	Constructor.
Nombre	OpenVideo(format: Format)
Descripción	Abre el video.
Nombre	GetFrame (position: double)
Descripción	Obtiene el frame o imagen que se encuentra la posición “position” del video.
Nombre	Source ()
Descripción	Obtiene la dirección física del video.
Nombre	VideoFormat ()
Descripción	Obtiene el formato del video.
Nombre	VideoSize ()
Descripción	Obtiene el tamaño del video.
Nombre	FrameRate ()
Descripción	Obtiene la frecuencia de imágenes por segundo.
Nombre	CountFrames ()
Descripción	Calcula el total de frames del video.
Nombre	CountFramesToProcess ()
Descripción	Calcula el total de frames que se van a procesar del video.

4.4.5 Clase ImageProcessor

Nombre	**ImageProcessor**
Tipo clase	Controladora
Para cada responsabilidad	
Nombre	ToGrayScale (image: Bitmap)
Descripción	Transforma una imagen a escala de grises.
Nombre	ToBinary (image: Bitmap)
Descripción	Binariza una imagen, utilizando un umbral, calculado por el método de Otsu.
Nombre	VerticalEdge (image: Bitmap)
Descripción	Se hallan los bordes de una imagen.
Nombre	OtsuThreshold (image: Bitmap)
Descripción	Obtiene un umbral de la imagen por el método de Otsu.
Nombre	InvertImage (image: Bitmap)
Descripción	Invierte una Imagen.
Nombre	Dilation (image: Bitmap, h: int, w: int)
Descripción	Hace la dilatación de una imagen teniendo en cuenta, una altura y un largo determinado.
Nombre	DepurateRegions (regiones: List<Region>, image: Bitmap)
Descripción	Eliminar regiones en la imagen que no cumplan con los requisitos geométricos establecidos para la suposición de texto.
Nombre	TextBox ([inout] regiones: List<Region>, image: Bitmap)
Descripción	Detecta las regiones candidatas a ser texto en una imagen.
Nombre	ConectedRegions(dimage: Bitmap)
Descripción	Determina las regiones conectadas.
Nombre	SubImage(region: Region, image: Bitmap)
Descripción	Obtiene dado una región y una imagen, la sub imagen que representa esa región.
Nombre	ImageofMatrix (matrix: int[][])
Descripción	Dada una matriz crea una imagen.

4.4.6 Clase ProcessManager

Nombre	**ProcessManager**
Tipo clase	Controladora
Atributo	Tipo
ocrprocess	CharacterRecognitionProcess
threadmanager	ThreadManager
imageprocessor	VideoProcessor
videoprocessor	ImageProcessor
videofiles	List<Video>
Para cada responsabilidad	
Nombre	ProgressIndexerEvent()
Descripción	Evento que se lanza cada vez que se procesa un fotograma del video.
Nombre	ProgressAnalizingRegion()
Descripción	Evento que se lanza cada vez que se procesa una región de un fotograma del video.
Nombre	ChangeThreadState()
Descripción	Evento que se lanza cada vez que cambia el estado de un proceso de un video.
Nombre	Video_Files ()
Descripción	Obtiene lista de videos en proceso.
Nombre	AddVideo (filename: string, url: string)
Descripción	Adiciona un video a la lista de videos en proceso.
Nombre	DeleteVideo (filename: string)
Descripción	Elimina un video de la lista de procesos dado su id.
Nombre	ProcesingVideo (filename: object)
Descripción	Se encarga de realizar el proceso completo al video hasta extraer el texto.
Nombre	StartProcess (id: string)
Descripción	Inicia el hilo que ejecuta todo el procesamiento de un video dado su id.
Nombre	PauseProcess (id: string)
Descripción	Pone en pausa el hilo que ejecuta todo el procesamiento de un video dado su id.
Nombre	StopProcess (id: string)
Descripción	Para el hilo que ejecuta todo el procesamiento de un video dado su id.
Nombre	Dispose ()
Descripción	Libera todo los recursos que esté utilizando el objeto.

4.4.7 Clase VideoProcessor

Nombre	**CharacterRecognitionProcess**	
Tipo clase	Controladora	
Atributo		Tipo
ocr		IOCRNetwork
Para cada responsabilidad		
Nombre	CharacterRecognitionProcess()	
Descripción	Constructor.	
Nombre	ConvertRegionToASCII (region: Region)	
Descripción	Convierte los caracteres reconocidos en una porción de imagen en formato de texto, ASCII.	
Descripción	Obtiene la dirección física del video.	
Nombre	VideoFormat ()	
Descripción	Obtiene el formato del video.	

En este capitulo se ha mostrado los diagramas de clases de análisis y diseño, así como los diagramas de secuencia de cada realización de los casos de uso. Se han mostrado las relaciones existentes entre las clases entidades y las controladoras y se describieron cada una de ellas.

CAPÍTULO 5: IMPLEMENTACIÓN

En el flujo de trabajo de implementación se convierten las clases y objetos en ficheros fuente, binarios, ejecutables entre otros. La finalidad principal es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. Además todos los componentes, características y requisitos deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del producto.

En el presente capítulo se realiza una breve descripción de los componentes del sistema propuesto, además se define el diagrama de componentes, en el cual se expone la ubicación de cada componente y su relación con el resto. También se muestran los resultados del estudio de compatibilidad realizado con la herramienta MoMA, con el fin de una migración a la plataforma UNIX.

5.1 Descripción de los componentes.

El sistema cuenta con tres ensamblados (Cabanes 2007): VideoProcessing.dll, ImageProcessing.dll, OCRProcessing.dll; que se encargan de distribuir las funcionalidades necesarias para la realización del algoritmo propuesto.

VideoProcessing.dll permite llevar a cabo acciones sobre el video digital, tales como extraer datos, por ejemplo, frecuencia de imágenes por segundos, duración, tamaño, formato, entre otras características. Además permite la reproducción del video y la descomposición de este en imágenes o fotogramas.

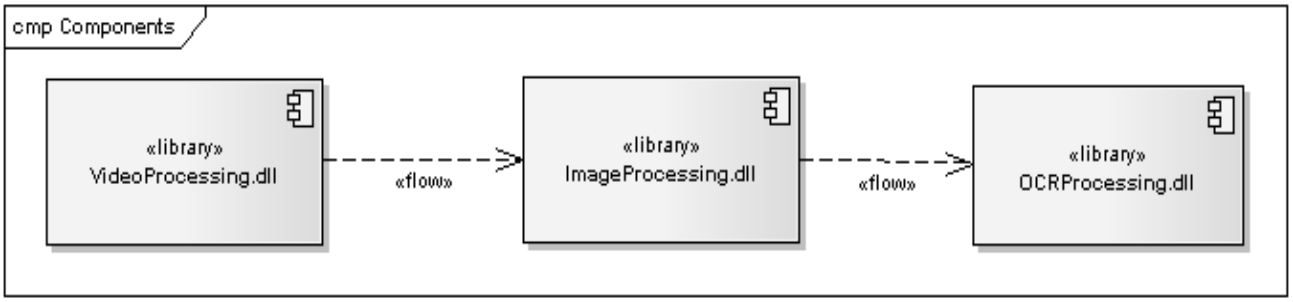
Por otro lado, ImageProcessing.dll realiza todo el trabajo sobre las imágenes extraídas del video, desde una simple transformación a escala de grises hasta lograr la detección de regiones candidatas a texto.

El componente OCRProcessing.dll ayuda en la última parte del proceso, es el encargado de: dadas ciertas regiones o partes de imágenes, lograr la extracción de del el texto, en formato texto (LI 2000). Para esto se utiliza, por las razones dadas en capítulos anteriores el módulo **LEADTOOLS OCR** (Danysoft n.d.)**.**

5.2 Diagrama de componentes.

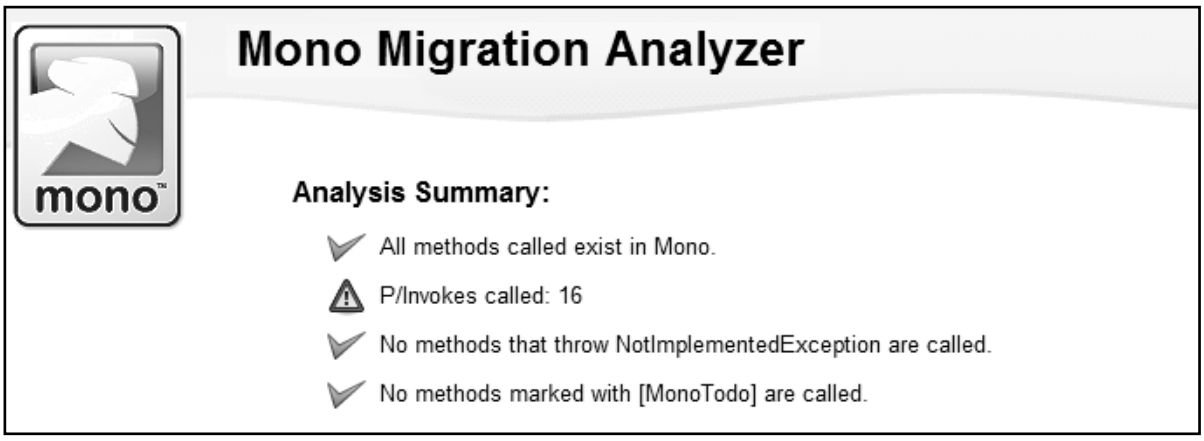
Los diagramas de componentes definen clases, artefactos, recursos y otros elementos de bajo nivel que se unen para conformar componentes que funcionan al más alto nivel. Todos ellos son artefactos del software compilado, que se convierten en esenciales para el funcionamiento del sistema informático. Estos trabajan acoplados, brindando el comportamiento que se requiere para el cumplimiento de los requerimientos antes expuestos en capítulos anteriores.

Los componentes se encuentran repartidos según plantea nuestra arquitectura, descrita en el capítulo anterior.



5.3 Estudio de compatibilidad con la plataforma libre.

Los estudios realizados con la herramienta MoMA (Mono Migration Analyzer), arrojan los siguientes resultados:



Detalles de los resultados:

ImageProcessing.dll

No Issues Found

OCRProcessing.dll

No Issues Found

VideoProcessing.dll

P/Invokes into native code

Calling Method	P/Invoke Method	External DLL
Class VideoFile.Win32:		
int AVISaveOptions(IntPtr, Win32/AVICOMPRESSOPTIONS&, IntPtr)	int AVISaveOptions(IntPtr, int, int, IntPtr[], IntPtr[])	avifil32.dll
int AVISaveOptions(IntPtr, Win32/AVICOMPRESSOPTIONS&, IntPtr)	int AVISaveOptionsFree(int, IntPtr[])	avifil32.dll
Class VideoFile.AVIReader:		
void Dispose(bool)	void AVIFileExit()	avifil32.dll
void Open(string)	int AVIFileOpen(IntPtr&, string, Win32/OpenFileMode, IntPtr)	avifil32.dll
void Open(string)	int AVIFileGetStream(IntPtr, IntPtr&, int, int)	avifil32.dll
void Open(string)	int AVIStreamInfo(IntPtr, Win32/AVISTREAMINFO&, int)	avifil32.dll
void Open(string)	IntPtr AVIStreamGetFrameOpen(IntPtr, Win32/BITMAPINFOHEADER&)	avifil32.dll
void Open(string)	IntPtr AVIStreamGetFrameOpen(IntPtr, Win32/BITMAPINFOHEADER&)	avifil32.dll
void Close()	int AVIStreamGetFrameClose(IntPtr)	avifil32.dll
void Close()	int AVIStreamRelease(IntPtr)	avifil32.dll
void Close()	int AVIFileRelease(IntPtr)	avifil32.dll
Bitmap GetFrame(double)	IntPtr AVIStreamGetFrame(IntPtr, int)	avifil32.dll
Bitmap GetFrame(double)	int memcpy(int, int, int)	ntdll.dll
Bitmap GetFrame(double)	int memcpy(int, int, int)	ntdll.dll
Bitmap GetFrame(double)	int memcpy(int, int, int)	ntdll.dll
void .ctor()	void AVIFileInit()	avifil32.dll

Como se observa todos los métodos existentes en los componentes del sistema se encuentran incluidos en la especificación de la versión 1.2.3 de Mono. Para los siguientes ensamblados no se han obtenido incompatibilidades para su migración:

Nombre del Ensamblado	Descripción
ImageProcessing.dll	En este ensamblado se realiza el procesamiento correspondiente a las imágenes.
OCRProcessing.dll	En este ensamblado se lleva a cabo la extracción del texto.

Para el ensamblado encargado del procesamiento del video, VideoProcessing.dll, se detectó un problema de compatibilidad, originado por el uso de la interfaz de programación de aplicación (API**,** por sus siglas en inglés) que se usa para la descomposición y reproducción del video: avifil32.dll.

Pero esto tampoco representa un problema, ya que en caso de realizar una migración del sistema hacia UNIX se usaría el MPlayer (MPLayer 2007), que puede ser fácilmente manejado desde C#

(Carballude 2008), y para integrarlo solo se tendría que implementar la interfaz “IVideo”, existente en el componente VideoProcessing.dll.

En este capítulo se realizó una breve descripción de los componentes del sistema propuesto, definiendo el diagrama de componentes, donde se le otorga la ubicación de cada componente y su relación con el resto. Además se muestran los resultados del estudio de compatibilidad realizado con la herramienta MoMA, donde se demuestra la posibilidad de una migración del sistema propuesto hacia UNIX.

CONCLUSIONES

Durante la presente investigación se desarrolló un sistema que permite un reconocimiento robusto de texto en secuencias de video, basado en detección de bordes, que tiene gran efectividad; logrando un índice de precisión de 81,3 % y extrayendo además el 95.1% del texto presente en los fotogramas. El tiempo promedio de procesamiento de 191.3 milisegundos por fotogramas es un aspecto fundamental. El sistema puede llevar a cabo el procesamiento concurrente de múltiples videos, que pueden estar en diferentes dominios comprimidos como MPEG-2, XVID, entre otros.

Se definió una arquitectura centrada en el flujo de datos, implementando el patrón “Filtros y Tuberías”, acorde a las necesidades operacionales, posibilitando la implementación de un sistema con alto rendimiento, extensibilidad, flexibilidad y reusabilidad del código.

Mediante el uso de Mono Migration Analyzer (MoMA), herramienta que ayuda a identificar diferentes problemas que se tengan al migrar las aplicaciones del Framework .NET al Framework de Mono, se logra demostrar la posibilidad de una migración del sistema en la plataforma libre.

La aplicación posibilita extraer la mayor parte del texto que se encuentra incrustado en los fotogramas de videos digitales. Esta información es útil en muchos sentidos y podría utilizarse para hacer búsquedas por contenido sobre un repositorio de videos procesados.

RECOMENDACIONES

Los autores de la presente investigación recomiendan:

- Continuar el estudio de las tendencias actuales en el campo de la detección y extracción de texto en videos digitales.
- Mejorar el método propuesto de detección y extracción de texto, en aras de obtener resultados superiores en cuanto a índice de precisión y detección.
- Continuar con la optimización del sistema para lograr mejoras en el tiempo de procesamiento así como en el uso de los recursos disponibles.
- Realizar la migración del sistema hacia plataformas libres.

REFERENCIAS BIBLIOGRÁFICAS

[1] ABBYY Software. «Sistema de reconocimiento óptico de caracteres ABBYY® FineReader.» 2005.

[2] Acharya, Tinku, y Ajoy K Ray. *Image Processing, Principles and Applications.* Wiley-Interscience, 2005.

[3] Anthimopoulos, Marios, Basilis Gatos, y Ioannis Pratikakis. «MULTIRESOLUTION TEXT DETECTION IN VIDEO FRAMES.» *International Conference on Computer Vision Theory and Applications*, 2006.

[4] Arora, Geetanjali, Nitin Pandey, y Balacubramadiam Aiswamy. Anaya Multimedia, 2002.

[5] Bimbo, Alberto del. «Bases de Datos de Imagenes y Videos.» Ciudad de la Habana, 2005.

[6] «Biometric Glossary.» *Resonance Pub The Portal to Science Engenieering and Technology*. 2008. http://www.resonancepub.com/biometricgl.htm.

[7] Cabanes, Nacho. *Diccionario de informática.* 04 de 02 de 2007. http://www.nachocabanes.com/diccio/ndic.php (último acceso: 04 de 03 de 2008).

[8] Carballude. «La playina del norte.» *La playina del norte.* 20 de 2 de 2008. http://www.carballude.es/Blog/?p=179 (último acceso: 5 de 3 de 2008).

[9] Corsi, Miguel. «Robin Good.» *Robin Good.* 2 de Agosto de 2007. http://www.masternewmedia.org/es/index.html (último acceso: 22 de Febrero de 2008).

[10] Cortés, Daniel Márquez. «Detección, extracción y reconocimiento de rótulos de texto en secuencias de video en dominio comprimido.» Tesis de Pregrado, 2005.

[11] Danysoft. *Haciendo visible lo invisible.* http://www.codegear-shop.com/epages/codegear-shop_com.sf/es_ES/?ObjectPath=/Shops/codegear-shop.com (último acceso: 04 de 03 de 2008).

[12] Gonzalez, Rafael C, y Richard E Woods. *Digital Image Processing.* 2002.

[13] Hur, Yongmin. «Test Data Compression Using a Hybrid Run-Length Code Method.» *Oxford Journals*, 2005.

[14] Icaza, Manuel. «Mono:la nueva plataforma de desarrollo para Unix, Linux y Mac OS.» *UOC _ La Universitat Oberta de Catalunya.* 2004. http://www.uoc.edu/dt/esp/deicaza0904.html.

[15] *IngenieroSoftware.* http://www.ingenierosoftware.com/analisisydiseno/uml.php (último acceso: 10 de 12 de 2008).

[16] Jacobson, Ivar, Grady Booch, y James Rumbaugh. *El Proceso Unificado de Desarrollo.* 2000.

[17] LI, Ning. «An Implementation of OCR System Based on Skeleton Matching.» 2000.

[18] Microsoft. «Office Online.» *Office Online.* Microsoft Corporation. 2008. http://office.microsoft.com/es-hn/help/CH010001183082.aspx (último acceso: 10 de 3 de 2008).

[19] Molina, Rafael. *Introducción al procesamiento y análisis de imágenes digitales.* Granada, 1998.

[20] Molpeceres, Alberto. «Procesos de Desarrollo RUP, XP y FDD.» 15 de 12 de 2002. http://www.willydev.net/descargas/articulos/general/cualxpfddrup.PDF (último acceso: 9 de 2 de 2008).

[21] *MPLayer.* 07 de 10 de 2007. http://www.mplayerhq.hu/design7/news-es.html (último acceso: 05 de 03 de 2008).

[22] «MSDN.» *Visual Source Safe.* http://msdn2.microsoft.com/es-es/library/3h0544kx(VS.80).aspx (último acceso: 10 de febrero de 2008).

[23] S. Morse, Bryan. «Thresholding.» 2002.

[24] Sánchez, Alfredo. «Paradigmas y estilos de interacción.» *Interacción Humano-Computadora.* 2007.

[25] Sazaklis, George M. «Geometric Methods for Optical Character Recognition.» 2002.

[26] Schmuller, Joseph. *Aprendiendo UML en 24 horas.* 2000.

[27] Wu, Victor, R Manmatha, y Edward Riseman. «An Automatic System to Detect and Recognize Text in Images.» 1997.

[28] Yingzi, Du, y Chang Chein-I. «Automated system for text detection in individual video images.» *Journal of Electronic Imaging,* 2003.

BIBLIOGRAFIA

[1] ABBYY Software. «Sistema de reconocimiento óptico de caracteres ABBYY® FineReader.» 2005.

[2] Acharya, Tinku, y Ajoy K Ray. *Image Processing, Principles and Applications.* Wiley-Interscience, 2005.

[3] Antani, S, D Crandal, y R Kasturi. «Robust Extraction of Text in Video.» 2000.

[4] Anthimopoulos, Marios, Basilis Gatos, y Ioannis Pratikakis. «MULTIRESOLUTION TEXT DETECTION IN VIDEO FRAMES.» *International Conference on Computer Vision Theory and Applications*, 2006.

[5] Arora, Geetanjali, Nitin Pandey, y Balacubramadiam Aiswamy. Anaya Multimedia, 2002.

[6] Assfalg, Jurgen, Marco Bertini, Carlo Colombo, Alberto Del Bimbo, y Walter Nunziati. «Semantic annotation of soccer videos:automatic highlights identification.» *Computer Vision and Image Understanding*, 2003.

[7] Bimbo, Alberto del. «Bases de Datos de Imagenes y Videos.» Ciudad de la Habana, 2005.

[8] «Biometric Glossary.» *Resonance Pub The Portal to Science Engenieering and Technology.* 2008. http://www.resonancepub.com/biometricgl.htm.

[9] Cabanes, Nacho. *Diccionario de informática.* 04 de 02 de 2007. http://www.nachocabanes.com/diccio/ndic.php (último acceso: 04 de 03 de 2008).

[10] Carballude. «La playina del norte.» *La playina del norte.* 20 de 2 de 2008. http://www.carballude.es/Blog/?p=179 (último acceso: 5 de 3 de 2008).

[11] Chen, Datong, Jean Marc Odobez, y Herve Bourlard. «Text detection and recognition in images and video frames.» 2003.

[12] Clark, P, y M Mirmehdi. «Combining Statistical Measures to Find Image Text Regions.» 2004.

[13] Clark, P, y M Mirmehdi. «Location and Recovery of Text on Oriented Surfaces.» 2004.

[14] Corsi, Miguel. «Robin Good.» *Robin Good.* 2 de Agosto de 2007. http://www.masternewmedia.org/es/index.html (último acceso: 22 de Febrero de 2008).

[15] Cortés, Daniel Márquez. «Detección, extracción y reconocimiento de rótulos de texto en secuencias de video en dominio comprimido.» Tesis de Pregrado, 2005.

[16] *Cursos Online.* 10 de 2 de 2008. http://www.adrformacion.com/curso/puntonet/leccion3/EntornoDesarrolloIntegrado.htm.

[17] «DANYSOFT.» *DANYSOFT.* DANYSOFT. 20 de 1 de 2007. http://www.danysoft.com/bol/070120lt.htm (último acceso: 10 de 3 de 2008).

[18] Danysoft. *Haciendo visible lo invisible.* http://www.codegear-shop.com/epages/codegear-shop_com.sf/es_ES/?ObjectPath=/Shops/codegear-shop.com (último acceso: 04 de 03 de 2008).

[19] Gatos, B, I Pratikakis, y S J Perantonis. «Adaptive degraded document image binarization.» *Pattern Recognition Society*, 2005.

[20] Gonzalez, Rafael C, y Richard E Woods. *Digital Image Processing.* 2002.

[21] Gu, Lifang. «Text Detection and Extraction in MPEG Video Sequences.» 2001.

[22] Guan, Ling, Sun Yuan Kung, y Jan Larsen. *MULTIMEDIA IMAGE and VIDEO PROCESSING.* New York: CRC Press, 2001.

[23] Hove, Lars Jacob. «Improving Content Based Image Retrieval Systems with a Thesaurus for Shapes.» Tesis maestria, 2004.

[24] Hua, Xian Sheng, Xiang Rong Chen, Liu Wenyin, y Hong Jiang Zhang. «Automatic Location of Text in Video Frames .» 2005.

[25] Hua, Xian Sheng, Yin Pei, y Hong Jiang Zhang. «EFFICIENT VIDEO TEXT RECOGNITION USING MULTIPLE FRAME INTEGRATION.» 2005.

[26] Hur, Yongmin. «Test Data Compression Using a Hybrid Run-Length Code Method.» *Oxford Journals*, 2005.

[27] Icaza, Manuel. «Mono:la nueva plataforma de desarrollo para Unix, Linux y Mac OS.» *UOC _ La Universitat Oberta de Catalunya.* 2004. http://www.uoc.edu/dt/esp/deicaza0904.html.

[28] *IngenieroSoftware.* http://www.ingenierosoftware.com/analisisydiseno/uml.php (último acceso: 10 de 12 de 2008).

[29] Jacobson, Ivar, Grady Booch, y James Rumbaugh. *El Proceso Unificado de Desarrollo.* 2000.

[30] Jung, Keechul, Kwang Kim, y Anil K Jain. «Text Information Extraction in Images and Video: A Survey.» 2004.

[31] León, Miriam, y Antoni Gasull. «TEXT DETECTION IN IMAGES AND VIDEO SEQUENCES.» 2004.

[32] Li, Huiping, David Doermann, y Omid Kia. «Automatic Text Detection and Traking in Digital Video.» 2002.

[33] LI, Ning. «An Implementation of OCR System Based on Skeleton Matching.» 2000.

[34] Lienhart, Rainer, y Axel Wernicke. «Localizing and Segmenting Text in Images and Videos.» *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY,* 2002.

[35] Lienhart, Rainer, y Wolfgang Effelsberg. «Automatic Text Segmentation and Text Recognition for Video Indexing.» *ACM/Springer Multimedia Systems Magazine,* 1998.

[36] Microsoft. «Office Online.» *Office Online.* Microsoft Corporation. 2008. http://office.microsoft.com/es-hn/help/CH010001183082.aspx (último acceso: 10 de 3 de 2008).

[37] Molina, Rafael. *Introducción al procesamiento y análisis de imágenes digitales.* Granada, 1998.

[38] Molpeceres, Alberto. «Procesos de Desarrollo RUP, XP y FDD.» 15 de 12 de 2002. http://www.willydev.net/descargas/articulos/general/cualxpfddrup.PDF (último acceso: 9 de 2 de 2008).

[39] *MPLayer.* 07 de 10 de 2007. http://www.mplayerhq.hu/design7/news-es.html (último acceso: 05 de 03 de 2008).

[40] «MSDN.» *Visual Source Safe.* http://msdn2.microsoft.com/es-es/library/3h0544kx(VS.80).aspx (último acceso: 10 de febrero de 2008).

[41] Pastor Gadea, Moises. «Aportaciones al Reconocimiento Automático de texto Manuscrito.» Tesis doctoral, Valencia, 2007.

[42] Russ, Jhon C. *The Image Processing Handbook.* Raleigh: CRC Press, 2002.

[43] S. Morse, Bryan. «Thresholding.» 2002.

[44] Sánchez, Alfredo. «Paradigmas y estilos de interacción.» *Interacción Humano-Computadora.* 2007.

[45] Sazaklis, George M. «Geometric Methods for Optical Character Recognition.» 2002.

[46] Schmuller, Joseph. *Aprendiendo UML en 24 horas.* 2000.

[47] Shen, Huiying, y James Coughlan. «Grouping Using Factor Graphs: an Approach for Finding Text with a Camera Phone.» 2002.

[48] Shijian, Lu, y Tan Chew Lim. «Binarization of Badly Illuminated Document Images through Shading Estimation and Compensation.» 2006.

[49] Tekinalp, Serhat. «DETECTING AND RECOGNIZING TEXT FROM VIDEO FRAMES.» Tesis maestria, 2002.

[50] Wolf, Christian Wolf, y Jean Michel Jolion. «Extraction and Recognition of Artificial Text in Multimedia Documents.» 2003.

[51] Wolf, Christian, y Jean Michel Jolion. «Model based text detection in images and videos: a learning approach.» 2003.

[52] Wu, Victor, R Manmatha, y Edward Riseman. «An Automatic System to Detect and Recognize Text in Images.» 1997.

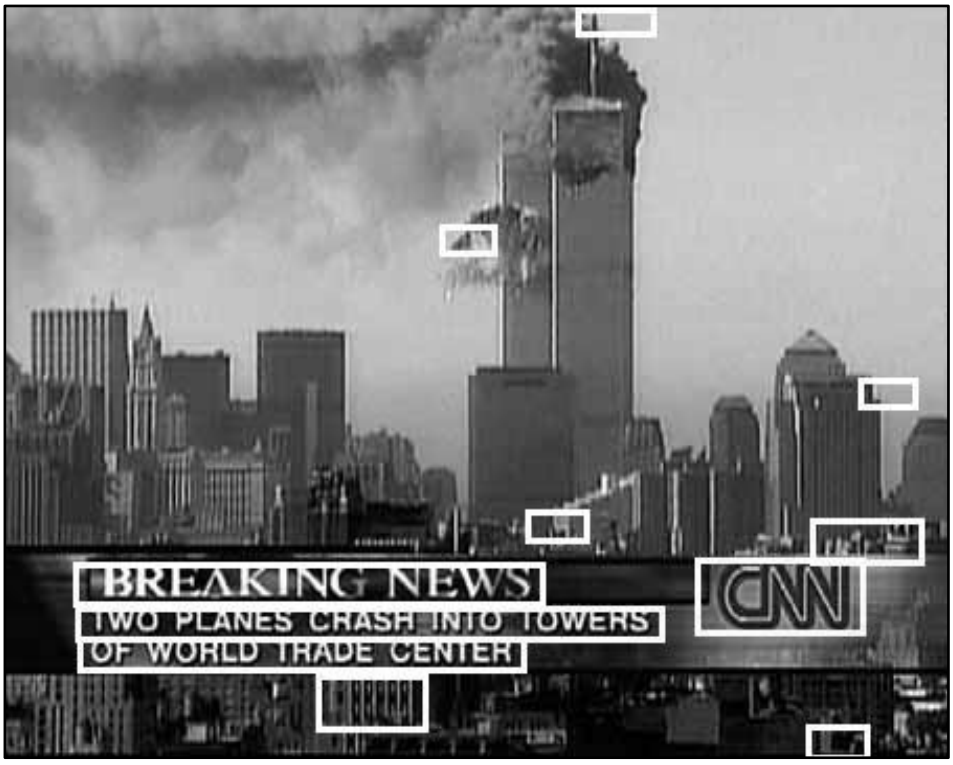
[53] Yingzi, Du, y Chang Chein-I. «Automated system for text detection in individual video images.» *Journal of Electronic Imaging*, 2003.

[54] Zhang, Yu Jin. *Advances in Image and Video Segmentation.* Beijing: IRM Press, 2006.

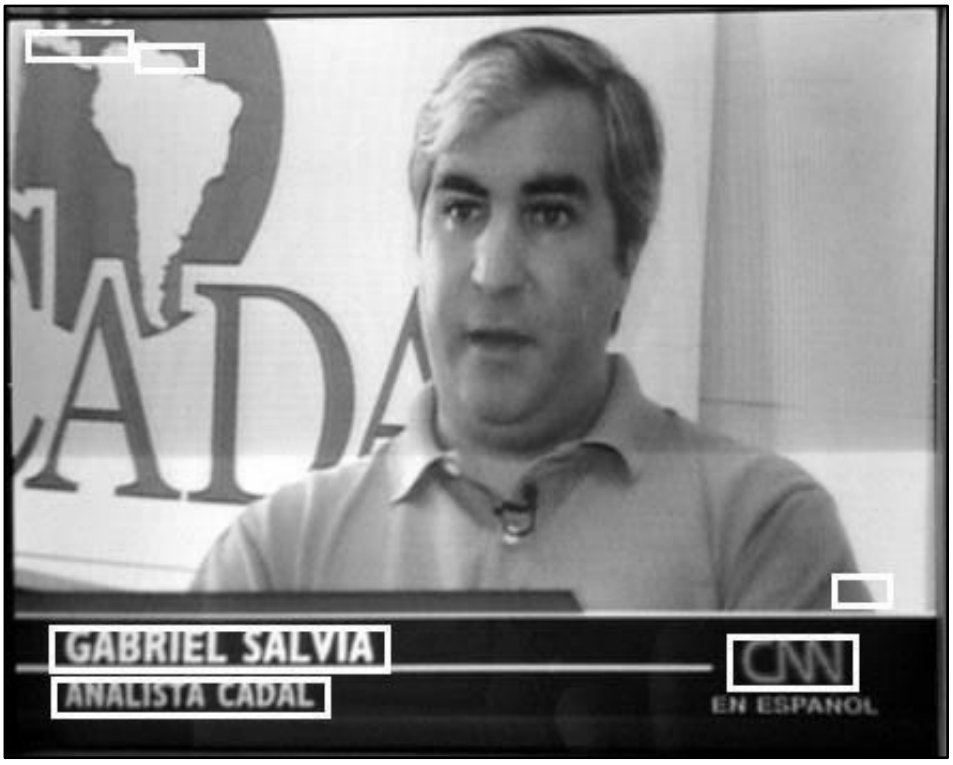
ANEXOS

ANEXO 1. Resultados del método de detección de texto propuesto.


CNN LIVE
Deadly Earthquake
Millions left homeless
and injured
Later
this
hour
Headlines
Pakistan earthquake measures 7.6 on the Richter scale


BREAKING NEWS
TWO PLANES CRASH INTO TOWERS
OF WORLD TRADE CENTER
CNN


imageGenerator.net
BREAKING NEWS

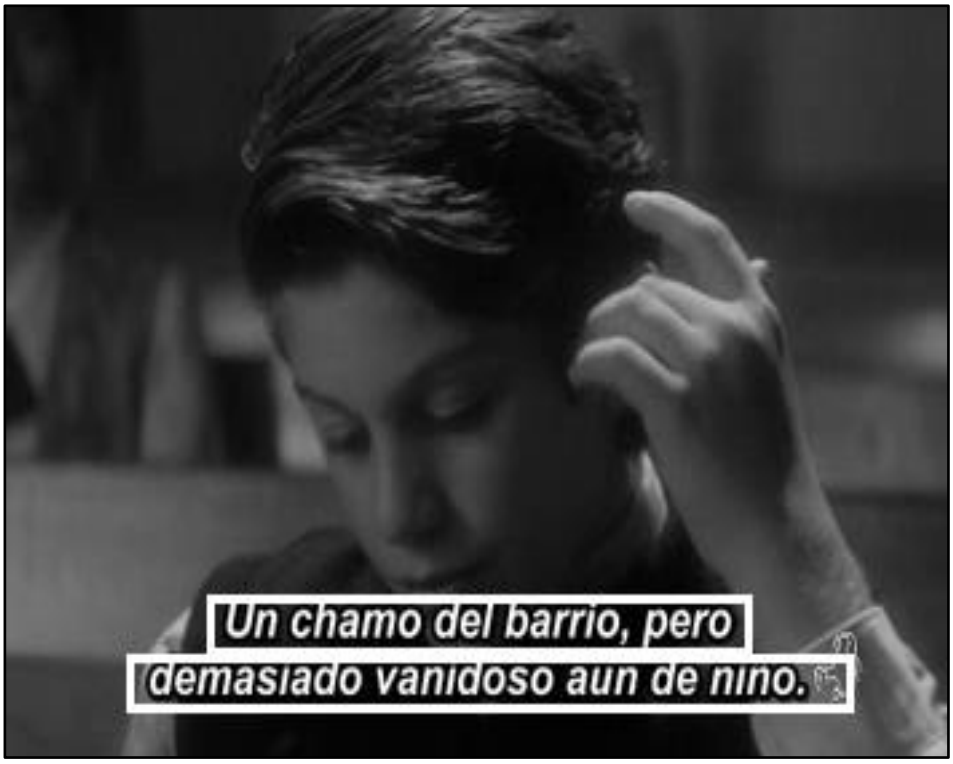

GABRIEL SALVIA
ANALISTA CADAL
CNN
EN ESPANOL


NEW LINE CINEMA
A Time Warner Company


El prisionero está
en la puerta del sur.
4


DIRECTOR OF PHOTOGRAPHY
MARC RESHOVSKY

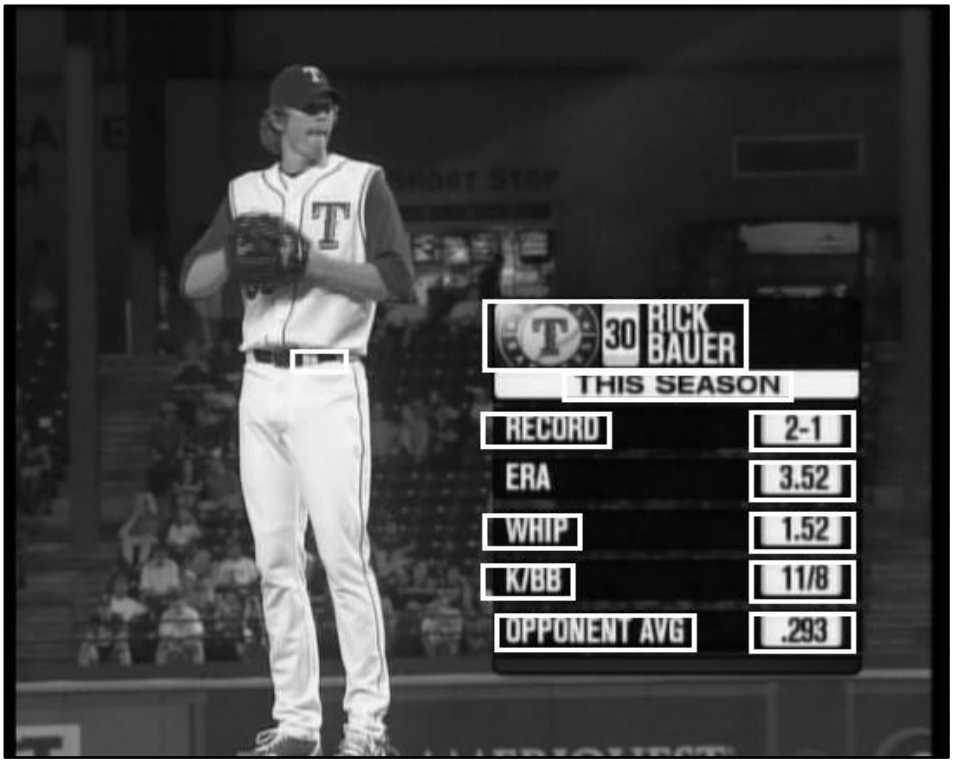

MOST
WANTED
EL MAS BUSCADO


Un chamo del barrio, pero
demasiado vanidoso aun de nino.


Este uniforme es un truco. Tenia que
salir para usar ellos el boca-de-campanilla


GOLTV
SANCHEZ RAMOS
CUARTO ÁRBITRO
ESQUERDO RODRIGUEZ
ASISTENTE 2
AGUILAR RODRIGUEZ
ASISTENTE 1
ÁLVAREZ IZQUIERDO
ÁRBITRO PRINCIPAL


1°
00:22
MAD
0
VIL
0
GOLTV
GOL PREMIUM ENGLISH
PHIL SCHOEN / RAY HUDSON


30 RICK BAUER
THIS SEASON
RECORD 2-1
ERA 3.52
WHIP 1.52
K/BB 11/8
OPPONENT AVG .293


LAA 7 TEX 6 ▲9TH 0 OUT 3-2 FSN WEST


LAA 7 TEX 6 ▲9TH
0 OUT
1-2
FSN WEST
22 KENDRY MORALES
1 FOR 4 TONIGHT
2-RUN HOME RUN IN 3RD


00:48
MAD 0
VIL 0
GOLTV
PARTIDO LIGA 06/07
REAL MADRID 0 0 VILLARREAL

GLOSARIO

Término	Definición
Video	Sucesión de imágenes presentadas a cierta frecuencia. La fluidez de un video se caracteriza por el número de imágenes por segundo (frecuencia de cuadros), expresado en FPS (cuadros por segundo). Generalmente está acompañado de sonido, es decir, datos de audio.
Video Digital	Sucesión de imágenes digitales. Dado que estas imágenes digitales o mapas de bits se muestran a una frecuencia determinada, es posible saber la frecuencia de bytes x segundos, es decir, el número de bytes mostrados (o transferidos) por unidad de tiempo. Posee además datos de audio, entre otros.
Imagen Digital	Mapa de bits. Representación binaria en la cual un bit o conjunto de bits corresponde a alguna parte de un objeto como una imagen o fuente. Por ejemplo, en sistemas monocromáticos, un bit en el mapa de bits representa un pixel en pantalla. Para la escala de grises o color, varios bits en el mapa de bits representan un pixel o grupo de pixeles. El término también puede hacer referencia al área de memoria que contiene el mapa de bits.
Bit	Es la unidad de información más pequeña. Puede tener sólo dos valores o estados: 0 o 1, encendido o apagado. La combinación de estos valores es la base de la informática, ya que los circuitos internos del ordenador sólo son capaces de detectar si la corriente llega o no llega (O o 1). Su nombre proviene de la contracción de las palabras «binary» y «digit» (dígito binario).
Fotograma	(en inglés, frame) Se le denomina a cada una de las imágenes impresas químicamente en la tira de celuloide del cinematógrafo, la película fotográfica o a cada una de las imágenes individuales captadas por cámaras de video y registradas analógica o digitalmente.
Segmentación	Técnica que divide la imagen digital en un conjunto de regiones **R**, donde los pixeles contenidos en la región $\mathbf{R_i}$ comparten atributos similares (Gonzalez and Woods 2002).
Detección de discontinuidades	La detección de discontinuidades es, esencialmente, una operación de detección de los cambios locales significativos en la intensidad de la imagen (Molina 1998).

Dilatación	La dilatación es la transformación morfológica que combina dos vectores utilizando la suma. Si A y B son conjuntos de un n-espacio E^n, con elementos a= (a_1,…, a_n) y b= (b_1,…, b_n), respectivamente, siendo ambos n-uplas, entonces la dilatación de A por B es el conjunto de todos los posibles vectores que son suma de pares de elementos, uno de A y otro de B (Molina 1998).
RUP	El Proceso Unificado de Desarrollo (RUP) es un proceso de desarrollo de software y un marco de trabajo genérico, que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organización, diferentes niveles de aptitud y diferentes tamaños de proyecto (Jacobson, Booch and Rumbaugh 2000).
UML	El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un software (Schmuller 2000).
CASE	Ingeniería de software asistida por computadora, por sus siglas en idioma Inglés.
Requisito funcional	Capacidad o condición que el sistema debe cumplir (Jacobson, Booch and Rumbaugh 2000).
Requisito no funcional	Propiedad o cualidad que el sistema debe tener (Jacobson, Booch and Rumbaugh 2000).
Clase	Estructura de datos empleada para crear objetos (Arora, Aiaswamy y Pandey 2002).
Ensamblado	Estructura lógica que contiene código compilado para la arquitectura .NET (Arora, Pandey and Aiswamy 2002).
Mono	Mono es la plataforma de desarrollo de software libre basada en .NET que permite a los desarrolladores de software construir aplicaciones GNU/Linux y multiplataforma con una productividad sin precedentes (Icaza 2004).
Trabajador	Puesto que debe ser asignado a una persona o equipo, y que requiere responsabilidades y habilidades como realizar actividades o desarrollar determinados artefactos (Jacobson, Booch and Rumbaugh 2000).
Falso positivo	Suceso que indica el cumplimiento de una condición en un individuo, cuando el individuo no la cumple (Biometric Glossary 2008).
Falso negativo	Suceso que indica el no cumplimiento de una condición en un individuo, cuando el individuo la cumple (Biometric Glossary 2008).