



Universidad de las Ciencias Informáticas

Facultad 6

*Título: Herramienta para la búsqueda y
reemplazo de logo en el subsistema de
transmisión en la Plataforma de Integración*

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

Autor: Liuben Alazales Capetillo

Tutora: Ing. Ekaterina A. Ramírez Ramírez

Co Tutor: Ing. Reinier Pupo Ruíz

Ciudad de La Habana, junio 2012

“Año 54 de la Revolución”

*“No es que queremos vivir así, es que
queremos vivir aquí”*

Amparo García.

Dedicatoria

Dedicatoria

A mis padres por estar los momentos cumbre de mi vida y por brindarme la posibilidad de realizar mis sueños.

A mis hermanas y hermano por dejarme ser ese ejemplo de hermano mayor.

A toda mi familia.

A todas mis amistades, que nunca dejaron de creer en mí.

A ustedes, dedico mis triunfos.

Agradecimientos

Quisiera darle las gracias:

A aquellas personas que han compartido conmigo todo estos años, el piquete de “Los Tíezos”, “Los Canecas” y a mis compañeros del aula, con los que he compartido innumerables batallas.

A mi familia, principalmente mis padres que siempre tuvieron esa confianza en mí y en especial a mi hermana Lisnory por ser mi ejemplo de consagración y optimismo, aunque nunca se lo haya dicho.

Gracias a mi tutora por ayudarme a ser el profesional en que me convierto hoy.

Líuben

Declaración de Autoría

Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas y a la Facultad 6 a hacer uso del mismo en su beneficio. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Liuben Alazales Capetillo

Ing. Ekaterina A. Ramírez Ramírez

Firma del Autor

Firma del Tutor

Resumen

En la actualidad la televisión juega un papel importante en la vida diaria de las personas, ya que constituye uno de los medios de comunicación más eficientes, efectivos y rápidos. Debido a la gran cantidad de instituciones que se dedican a este tipo de trabajo es necesario tener controlado la cantidad de canales que se transmiten a diario. Cada canal que es transmitido posee un logo, característica común que los diferencia unos de otros.

El Departamento de Señales Digitales de la Universidad de las Ciencias Informáticas ha sentido la necesidad de mantener identificados y controlados los canales televisivos que a diario se transmiten, debido a la variedad existente en nuestro país, este proceso se torna un poco engorroso a la hora de realizarlo manualmente.

El objetivo de la presente investigación es desarrollar una herramienta capaz de realizar el proceso de detección y reemplazo de los logos existentes en una media de forma automática. Recogiendo en ella la documentación referente a los resultados de estudios y análisis realizados a las técnicas de procesamiento de imagen digital, para obtener conocimiento acerca de los algoritmos de detección de objetos en una imagen. Así como también los estudios realizados para la determinación de las herramientas, lenguaje de programación y tecnologías para su desarrollo. Se describen también las funcionalidades que debe cumplir la herramienta así como la forma que estará estructurada y los Modelos de análisis y diseño de la misma. Se tendrá una descripción del algoritmo utilizado y de las pruebas realizadas al mismo para la evaluación de su calidad.

Palabras Claves

Detección, logo, media, televisión, transmisión.

Índice General

ÍNDICE DE GENERAL

INTRODUCCIÓN.....	1
CAPÍTULO 1. Fundamentación Teórica.....	6
1.1 Introducción.....	6
1.2 Conceptos asociados.....	6
1.3 Objeto de Estudio.....	8
1.3.1 Descripción General.....	8
1.4 Análisis de soluciones existentes.....	13
1.4.1 Videoma Broadcast Monitor (VBM).....	13
1.4.2 TdTarsys.....	13
1.5 Tendencias y Tecnologías para el desarrollo de software.....	14
1.5.1 RUP. Metodología de Desarrollo.....	14
1.5.2 Visual Paradigm. Herramienta de Modelado.....	17
1.5.3 C++. Lenguaje de Programación.....	17
1.5.4 Qt-Creator. Entorno de Desarrollo Integrado.....	18
1.5.5 Biblioteca OpenCV.....	18
1.5.6 Sistema Gestor de Base de Datos.....	19
1.6 Conclusiones.....	20
CAPÍTULO 2. Características del Sistema.....	21
2.1 Introducción.....	21
2.2 Sistema propuesto.....	21
2.3 Modelo de domino.....	21
2.4 Especificación de requisitos de software.....	22
2.4.1 Requisitos Funcionales.....	22
2.4.2 Requisitos no Funcionales.....	23
2.5 Caso de Uso del Sistema.....	24

Índice General

2.5.1 Actores.....	24
2.5.2 Diagramas de Casos de Uso.	24
2.5.3 Descripción de los Casos de Uso.	25
2.6 Conclusiones.....	26
CAPÍTULO 3. Análisis y Diseño del Sistema	27
3.1 Introducción.....	27
3.2 Patrón de Arquitectura.	27
3.3 Patrones de diseño.	27
3.4 Modelo de Análisis.....	28
3.4.1 Diagrama de Clases del Análisis.	29
3.4.2 Diagrama de Colaboración.....	30
3.5 Modelo del diseño.	31
3.6 Conclusiones.....	33
CAPÍTULO 4. Implementación y Prueba.....	34
4.1 Introducción.....	34
4.2 Estándares de Codificación.	34
4.3 Análisis del algoritmo	35
4.4 Modelo de Implementación.....	38
4.4.1 Diagrama de Componente.	38
4.4.2 Diagrama de Despliegue.	39
4.5 Pruebas de Software.	39
4.5.1 Plan de Prueba.	40
4.5.2 Prueba de Caja Blanca.	40
4.5.3 Diseño de casos de prueba.....	41
4.5.4 Prueba de Eficiencia.....	44
4.6 Conclusiones.....	52
CONCLUSIONES GENERALES.....	53

Índice General

RECOMENDACIONES.....	54
BIBLIOGRAFÍA REFERENCIADA	55
BIBLIOGRAFÍA	56
GLOSARIO DE TÉRMINOS	60

ÍNDICE DE FIGURAS

Figura 1: Logos (a).....	7
Figura 2: Logos (b).....	7
Figura 3: Logo original.	10
Figura 4: Logo en escala de gris.	10
Figura 5: Logo aplicando Binarización.	10
Figura 6: Fases de RUP (11).....	15
Figura 7: Diagrama del Modelo de Dominio.....	22
Figura 8: Diagramas de Casos de Uso.	24
Figura 9: Diagrama de Clase del Análisis Detección de Logo.....	29
Figura 10: Diagrama de Clase del Análisis Reemplazar Logo.	30
Figura 11: Diagrama de Colaboración Detección de Logo.	30
Figura 12: Diagrama de Colaboración Reemplazar Logo.	31
Figura 13: Diagrama de Clase del Diseño.	32
Figura 14: Conversión de imagen.	35
Figura 15: Binarización de imagen.	36
Figura 16: Multiplicación lógica.	37
Figura 17: Logo del video.	37
Figura 18: Imagen con rectángulo envolvente	38
Figura 19: Diagrama de Componentes.	39
Figura 20: Diagrama de Despliegue.....	39
Figura 21: Grafo de Flujo.	42
Figura 22: Imagen del video Evolution.	47
Figura 23: Imágenes del video Grimm.	48
Figura 24: Imágenes del video Sky.	49
Figura 25: Imágenes del video Day.....	50

ÍNDICE DE TABLAS

Tabla 1: RUP vs XP.	15
Tabla 2: Actores.	24
Tabla 3: Descripción del Caso de Uso Detección de Logo.	25
Tabla 4: Descripción del Caso de Uso Reemplazo de Logo.	26
Tabla 5: Resultado de los Casos de Prueba.	44
Tabla 6: Resultado del video Evolution	47
Tabla 7: Resultado del video Grimm.	48
Tabla 8: Resultado del video Sky.	49
Tabla 9: Resultados del video Day.	50
Tabla 10: Resultado de prueba (once videos).	51

INTRODUCCIÓN

El hombre a lo largo de la historia se ha abierto paso ante las diferentes adversidades a las que ha tenido que enfrentarse, caracterizándose por su forma de revolucionar todo aquello que lo rodea. Haciendo uso de sus conocimientos, en su necesidad de progresar ha creado innumerables objetos entre los que se encuentra la cámara fotográfica. Esta fue creada para poder capturar en una lámina de cartón una imagen o fotografía. La capacidad de capturar en imágenes ciertos hechos importantes de la realidad, la ha convertido en una herramienta indispensable para la ciencia, el arte, la publicidad y el periodismo. En cada campo y de acuerdo a la mirada de cada uno de los hacedores de su historia, la fotografía ha actuado como el difusor de ciertas ideologías y un agente significativo de primera mano. De la fotografía el hombre con su constante evolucionar saltó al video o cine donde se fusionaba un gran número de imágenes en movimiento, el sonido se le une más tarde para así crear una réplica exacta de lo que acontecía en un momento determinado de la historia de la humanidad.

Años más tarde el hombre con su revolucionar y sus deseos de mejorar llega a la producción de la televisión, que se desarrolló con los avances técnicos que permitieron la grabación de las señales de video y audio. Esto permitía la realización de programas grabados para una posterior emisión. El desarrollo de la televisión no se detuvo, también se implementaron sistemas de sonido mejorados así como la unificación a este elemento de filtros rojos, verdes y azules logrando emitir las primeras imágenes en color.

Poco después se comienzan a desarrollar equipos basados en la digitalización de señales de video y en la generación de señales digitales, naciendo así los efectos digitales en la televisión. A su vez con el desarrollo de la electrónica, que permite el montaje de salas de post-producción que combinados con otros elementos se pueden realizar programas más complejos.

La televisión digital (o DTV, de sus siglas en inglés: digital TV) se refiere al conjunto de tecnologías de transmisión y recepción de imagen y sonido, a través de señales digitales. En contraste con la televisión tradicional, que codifica los datos de manera analógica, la televisión digital codifica sus señales de forma binaria, habilitando así la posibilidad de crear vías de retorno entre consumidor y productor de contenidos, abriendo nuevas alternativas de crear aplicaciones interactivas, y la capacidad de transmitir varias señales en un mismo canal asignado, gracias a la diversidad de formatos existentes. Con la televisión digital se aumentaron los canales a transmitir así como la calidad de las señales a transmitir por los mismos.

Introducción

Con el desarrollo de las nuevas tecnologías y la llegada de la televisión digital comenzó una carrera competitiva entre las diferentes compañías que se dedicaron al tema de la transmisión, ya que esta empezó a jugar un papel importante en la vida del hombre, debido a que por medio de la televisión le llegaban las informaciones importantes del acontecer nacional y extranjero. La necesidad de que la audiencia reconociera sus productos propició la emisión del logotipo de las compañías junto con los materiales de la programación de estas.

El Departamento Señales Digitales perteneciente al centro GEYSED de la Facultad 6 en la Universidad de las Ciencias Informáticas (UCI), no se mantiene fuera del alcance de estos procesos que se llevan a cabo en el mundo, en dicho departamento surge un grupo de investigación, innovación y desarrollo llamado Plataforma de Integración (SIAV) que tiene como uno de sus objetivos permitir que la televisión universitaria digitalice sus media, este cuenta además con un componente para la monitorización y transmisión automática de los canales de televisión que se transmiten.

Debido al desarrollo de las comunicaciones y al número de canales con que cuenta el país se tiene un aumento del uso de los materiales audiovisuales en diferentes sectores de la sociedad. Esto ha provocado que la transmisión de canales juegue un papel significativo en estos sectores, ya que a través de este medio, la información llega con más rapidez y facilidad a las personas.

Se hace necesaria la monitorización y control de los canales televisivos que se transmiten a diario, con el objetivo de que a la población sobre la que está dirigida la transmisión de la media, sea capaz de elegir o distinguir los canales.

Sin una insignia que diferencie los canales no será posible una transmisión de calidad y del agrado para los televidentes, debido a que no tendrán claro hacia qué canal se sienten más atraídos o prefieren ver. De igual manera no existiría la forma de realizar una evaluación crítica y autocrítica de la programación que se lleva a la población por los canales que transmiten. Trayendo esto consigo que la transmisión, al no ser de buena calidad y la población al no tener claro que canal que transmite sus medias de preferencia busquen otro tipo de medios para suplir sus necesidades ya sea de información como de curiosidad a los cambios que ocurren en el mundo que los rodea.

Por lo que se piensa, que con el desarrollo de una herramienta que permita la representación de manera efectiva de los canales de transmisión de manera automática, se eliminará la acción de la mano del hombre que podría llevar a errores en la representación y así en la transmisión. Se ha hecho inevitable que la transmisión sea de manera eficiente sin errores, con la mayor calidad posible, es por

Introducción

ello que se ha puesto en desarrollo en el subsistema de transmisión esta herramienta capaz de realizar esta tarea. Esto agregará la característica de ser una solución propia de la Plataforma de Integración.

Teniendo en cuenta la situación anteriormente planteada se define como **problema a resolver**: ¿Cómo representar los canales en la transmisión de la Plataforma de Integración?

Como **objetivo general** se propone: Desarrollar una herramienta que permita la representación de los canales en transmisión.

Para dar solución en correspondencia al problema formulado el **objeto de estudio** es definido por: Tratamiento de imágenes en contenido audiovisual.

Enmarcando como **campo de acción**: proceso de transmisión de señales audiovisuales en la Plataforma de Integración.

Proponiendo por lo antes expuesto en la investigación como **idea a defender** que con el desarrollo de una herramienta que permita el correcto posicionamiento de logos en la transmisión de la Plataforma de Integración se logrará la representación de los canales durante la transmisión.

Para dar cumplimiento al objetivo propuesto se han trazado las siguientes **tareas de investigación**:

- Caracterizar el estado del arte referente a las diferentes herramientas o soluciones existentes que permitan analizar el funcionamiento de sistemas para el posicionamiento de logos en la transmisión de señales audiovisuales.
- Caracterizar las tendencias y tecnologías actuales más apropiadas para el desarrollo de la herramienta.
- Elaborar artefactos y documentación correspondiente al proceso de desarrollo del software.
 - Especificar los procesos de negocio.
 - Especificar requisitos de software (Funcionales, No Funcionales).
 - Elaborar el modelo del sistema.
 - Elaborar el modelo de análisis de sistema.
 - Elaborar el modelo de diseño del sistema.
 - Elaborar el modelo de implementación de sistema.
 - Elaborar el modelo de despliegue.
- Implementar herramienta.
- Validar la herramienta desarrollada.

Introducción

Durante el desarrollo de esta investigación, y para dar respuesta al problema planteado, se aplicarán diferentes **métodos científicos**. Estos permiten organizar el trabajo, lo cual posibilita el desarrollo de un amplio conocimiento referente al estado del arte tratado y su evolución en etapas determinadas.

Métodos Teóricos

Analítico-Sintético: Permitió analizar teorías y documentos sobre las herramientas de búsqueda y generación de logos para la posterior representación de los canales. Se emplea también para caracterizar soluciones similares que permitan aprovechar puntos en común y conceptos teóricos que sean de relevancia para la investigación.

Análisis Histórico-Lógico: Permitió el estudio de diferentes herramientas de búsqueda y generación de logos para la representación de los mismo en la transmisión de canales, cuándo fueron desarrollados, así como su evolución en el tiempo. Se realizaron comparaciones entre las herramientas encontradas lo que facilitó la obtención de elementos de interés para decidirse por una u otra para utilizar la óptima.

Modelación: Durante la investigación se utiliza para crear abstracciones que permitan comprender el funcionamiento del sistema mediante los modelos de implementación, modelos de análisis y diseño del sistema y modelo de despliegue.

Método Empírico

Observación: Consiste en la percepción directa del objeto de investigación. La observación investigativa es el instrumento universal del científico. La observación permite conocer la realidad mediante la percepción directa de los objetos y fenómenos.

Descripción de Capítulos

Capítulo 1: En este capítulo se analizan las diferentes técnicas para la detección de logos así como las metodologías y tecnologías a utilizar.

Capítulo 2: En este capítulo se hará referencia a las características que tendrá la herramienta a desarrollar, como es el caso de sus requisitos tanto funcionales como no funcionales y sus casos de uso.

Capítulo 3: En este capítulo se presentará la propuesta de solución de la herramienta, con la presentación de patrones de arquitectura y diseño, así como también la de los diagramas de diseño.

Introducción

Capítulo 4: En este capítulo se podrá observar temas relacionados con implementación de la herramienta así como los resultados de las pruebas que se le realizara para validar la misma.

Capítulo 1. Fundamentación Teórica

CAPÍTULO 1. Fundamentación Teórica

1.1 Introducción.

En el siguiente capítulo se abordan los temas referentes a las imágenes y videos digitales. Se brinda información del estado actual de las bibliotecas y aplicaciones que realizan detección y remplazo de logos en imágenes y videos digitales, así como las diferentes técnicas que pueden ser utilizadas para este proceso.

1.2 Conceptos asociados.

Como parte de la fundamentación teórica se hace necesario tener en cuenta algunos conceptos para la correcta comprensión de la investigación.

Imagen digital

Una imagen digital es una matriz de $M \times N$ elementos que se obtiene 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. (1)

Imagen digital es la representación bidimensional de una imagen empleando bits. De acuerdo a la resolución que presenta la imagen, estática o dinámica, se podrá hablar de gráfico rasterizado (o mapa de bits; representa una rejilla rectangular de píxeles o punto de color, la cual se puede visualizar en un monitor, en un papel o en cualquier otro dispositivo de representación que se emplee) o de gráfico vectorial (imagen producto de objetos geométricos independientes). (2)

Se puede resumir que la imagen digital no es más que la representación bidimensional de una imagen analógica después de haber pasado por un proceso de discretización. Dependiendo de su resolución es clasificada como imagen estática o dinámica.

Video digital

El video digital es un tipo de sistema de grabación de video que funciona usando una representación digital de la señal de video. Es el resultado de la combinación de secuencias de imágenes, audio, texto y movimiento que se obtienen por muestreo y cuantificación de la señal de video analógica. (3)

El video digital es la grabación de un video usando la representación de señales digitales y no analógicas, es la combinación de texto, movimiento, audio e imágenes.

Capítulo 1. Fundamentación Teórica

Logo

Distintivo o emblema conformado a partir de letras e imágenes, abreviaturas, entre las alternativas más comunes, peculiar y particular de una empresa, producto o marca determinada. (4)

El logo es la manera de representar a una empresa, producto o marca usando caracteres e imágenes sin hacer uso de su nombre completo, véase la **Figura 1 y 2**.



Figura 1: Logos (a)



Figura 2: Logos (b)

Capítulo 1. Fundamentación Teórica

Fotogramas

También denominados frames, constituyen la unidad básica de información y generalmente son filmados a una velocidad de 1/25 o 1/30 segundos. Básicamente consisten en imágenes estáticas cuya reproducción a la velocidad adecuada crean la sensación de movimiento característica de un video. (5)

1.3 Objeto de Estudio.

1.3.1 Descripción General.

Para la detección de logotipos mediante el tratamiento de imágenes en un video se pueden usar diferentes tipos de técnicas y métodos con el fin de poder identificar los canales y el audiovisual que está en transmisión.

Durante el tratamiento de las imágenes digitales se podrá obtener como resultado del proceso el realzar bordes: destacar los bordes que se localizan en una imagen, detectar bordes: detectar los píxeles donde se produce un cambio brusco en la función intensidad. A continuación se mostrarán diferentes tipos de técnicas que se utilizan con este fin.

1.3.1.1 Detección de Logo.

Los logos de televisión suelen ser fijos y comprenden muy pocos píxeles de la imagen, por ejemplo varios cientos de píxeles de la imagen a tamaño CIF en formato MPEG-1.

Con el fin de hacer pleno uso de la información de máscara, se lleva a cabo la detección de logotipo en el espacio sub-píxel. Se agranda la imagen al triple con uso de la interpolación bilineal. Dada una imagen, el algoritmo de OTSU se emplea para determinar automáticamente un umbral óptimo para binarizar la imagen. Este algoritmo es ejecutado por la maximización de la variable medida discriminante de una imagen en niveles de gris. Dado una imagen representada por L niveles de gris [1, 2, 3...L]. El número de píxeles en el nivel i es n_i , el número total de píxeles es N, el histograma se normaliza y la consideran como una distribución de probabilidad:

$$p_i = \frac{n_i}{N}$$

Al utilizar el momento acumulado cero $\omega(k) = \sum_{i=1}^k p_i$ y el momento acumulado primera $\mu(k) = \sum_{i=1}^k i p_i$ del histograma, el k umbral óptimo se puede obtener por el criterio discriminativo de la siguiente manera:

Capítulo 1. Fundamentación Teórica

$$\sigma^2(k^*) = \max_{1 \leq k \leq L} \left(\frac{(\mu(L)\omega(k) - \mu(k))^2}{\omega(k)(1 - \omega(k))} \right)$$

Los bordes de una imagen digital se pueden definir como transiciones entre dos regiones de niveles de gris significativamente distintos. Suministran una valiosa información sobre las fronteras de los objetos y puede ser utilizada para segmentar la imagen, reconocer objetos, etc. La mayoría de las técnicas para detectar bordes emplean operadores locales basados en distintas aproximaciones discretas de la primera y segunda derivada de los niveles de grises de la imagen. (6)

Operadores basadas en la primera derivada (Gradiente).

La derivada de una señal continua proporciona las variaciones locales con respecto a la variable, de forma que el valor de la derivada es mayor cuanto más rápidas son estas variaciones. En el caso de funciones bidimensionales $f(x, y)$, la derivada es un vector que apunta en la dirección de la máxima variación de $f(x, y)$ y cuyo módulo es proporcional a dicha variación. Este vector se denomina gradiente y se define:

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix}$$

$$Mag[\nabla f(x, y)] = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2}$$

$$\theta = \arctan \frac{\frac{\partial f(x, y)}{\partial x}}{\frac{\partial f(x, y)}{\partial y}}$$

En el caso bidimensional discreto, las distintas aproximaciones del operador gradiente se basan en diferencias entre los niveles de grises de la imagen. La derivada parcial de la función (gradiente de fila $GF(i, j)$) puede aproximarse por la diferencia de píxeles adyacentes de la misma fila.

$$\frac{\partial f(x, y)}{\partial x} \approx \nabla_x f(x, y) = f(x, y) - f(x - 1, y) [-1, 1]$$

La discretización del vector gradiente en el eje Y ($GC(i, j)$), será:

$$\frac{\partial f(x, y)}{\partial y} \approx \nabla_y f(x, y) = f(x, y) - f(x, y - 1) [-1, 1]$$

Capítulo 1. Fundamentación Teórica

El gradiente de la fila GF y de columna GC en cada punto se obtienen mediante la convolución de la imagen con las máscaras HF y HC, esto es:

$$GF(i, j) = F(i, j) \blacksquare HF(i, j)$$

$$GC(i, j) = F(i, j) \blacksquare HC(i, j)$$

La magnitud y orientación del vector gradiente suele aproximarse por la expresión:

$$G(i, j) = GF + GC \approx GF(i, j) + GC(i, j) \quad (7)$$

1.3.1.2 Binarización.

Es uno de los métodos de umbralización más simple entre las técnicas de segmentación, donde por medio de la generación de una imagen binaria se busca separar entre objetos y fondo. En la umbralización se transforma una imagen en escala de grises a binaria, según el valor de cada "pixel" es mayor o menor que el valor de un determinado umbral. Dado este valor se pondrá en cero los pixeles menores que el umbral y en uno aquellos que sean mayores o igual. En la **Figura 3, 4 y 5** se muestran ejemplos del proceso de binarización de una imagen desde su estado original.



Figura 3: Logo original.



Figura 4: Logo en escala de gris.



Figura 5: Logo aplicando Binarización.

1.3.1.3 Template Matching.

El método más sencillo para realizar esta tarea consiste en la comparación de plantillas. En consideración a la posición constante de un logotipo de televisión, se calcula el gradiente generalizado sobre una secuencia de imágenes para construir la plantilla correspondiente. La acumulación temporal

ayuda a reducir los ruidos de fondo y producir un contorno claro del logotipo de la TV. La plantilla final se obtiene por binarización de la imagen.

El número de fotogramas vecinos está estrechamente relacionado con el modelado de la plantilla. Marcos insuficientes no pueden eliminar los ruidos de los contenidos de fondo que podrían aumentar las falsas alarmas. Demasiados marcos tienden a desdibujar las características inherentes de un logotipo que podría aumentar los segmentos perdido. Por otra parte, el número de fotogramas adyacentes afecta a la resolución temporal del seguimiento.

Por lo tanto, un sistema de coincidencia de logotipo a dos niveles se propone que en ambos niveles, la existencia o ausencia del logotipo de la TV se decide por los criterios de coincidencia:

$$C(I, T) = \sum_{T(i,j)=1} (I(i,j))$$

Donde $I(i, j)$ es la imagen binaria derivada de los gradientes de marcos consiguientes, $T(i, j)$ es la plantilla correspondiente. Si $C(I, T) \geq Th^1$, el logotipo de TV existe, de lo contrario, está ausente. (6)

1.3.1.4 Algoritmo de Canny para la detección de bordes.

Uno de los métodos relacionados con la detección de bordes es el uso de la primera derivada, la que es usada por que toma el valor de cero en todas las regiones donde no varía la intensidad y tiene un valor constante en toda la transición de intensidad. Por tanto un cambio de intensidad se manifiesta como un cambio brusco en la primera derivada, característica que es usada para detectar un borde, y en la que se basa el algoritmo de Canny.

El algoritmo de Canny consiste en tres grandes pasos:

- Obtención del gradiente.
- Supresión no máxima.
- Histéresis de umbral.

Obtención del gradiente

Para la obtención del gradiente, lo primero que se realiza es la aplicación de un filtro gaussiano a la imagen original con el objetivo de suavizar la imagen y tratar de eliminar el posible ruido existente. Sin

¹ Threshold o umbral en español.

Capítulo 1. Fundamentación Teórica

embargo, se debe de tener cuidado de no realizar un suavizado excesivo, pues se podrían perder detalles de la imagen y provocar un pésimo resultado final. Este suavizado se obtiene promediando los valores de intensidad de los píxeles en el entorno de vecindad con una máscara de convolución de media cero y desviación estándar σ . Una vez que se suaviza la imagen, para cada píxel se obtiene la magnitud y módulo (orientación) del gradiente, obteniendo así dos imágenes.

Supresión no máxima (gradiente)

Las dos imágenes generadas en el paso anterior sirven de entrada para generar una imagen con los bordes adelgazados. El procedimiento es el siguiente: se consideran cuatro direcciones identificadas por las orientaciones de 0° , 45° , 90° y 135° con respecto al eje horizontal. Para cada píxel se encuentra la dirección que mejor se aproxime a la dirección del ángulo de gradiente. Posteriormente se observa si el valor de la magnitud de gradiente es más pequeño que al menos uno de sus dos vecinos en la dirección del ángulo obtenida en el paso anterior. De ser así se asigna el valor 0 a dicho píxel, en caso contrario se asigna el valor que tenga la magnitud del gradiente. La salida de este segundo paso es la imagen con los bordes adelgazados, después de la supresión no máxima de puntos de borde.

Histéresis de umbral (supresión no máxima)

La imagen obtenida en el paso anterior suele contener máximos locales creados por el ruido. Una solución para eliminar dicho ruido es la histéresis del umbral. El proceso consiste en tomar la imagen obtenida del paso anterior, tomar la orientación de los puntos de borde de la imagen y tomar dos umbrales, el primero más pequeño que el segundo. Para cada punto de la imagen se debe localizar el siguiente punto de borde no explorado que sea mayor al segundo umbral. A partir de dicho punto seguir las cadenas de máximos locales conectados en ambas direcciones perpendiculares a la normal del borde siempre que sean mayores al primer umbral. Así se marcan todos los puntos explorados y se almacena la lista de todos los puntos en el contorno conectado. Es así como en este paso se logra eliminar las uniones en forma de Y de los segmentos que confluyen en un punto. (8)

Después de haber realizado un amplio análisis sobre las diferentes técnicas y métodos que se pueden usar para el procesamiento de imágenes sobre un contenido audiovisual para la detección de logos, lanzando como resultado que la mejor opción para la detección de logos es haciendo uso de la Binarización.

Capítulo 1. Fundamentación Teórica

1.4 Análisis de soluciones existentes.

En la actualidad el mundo del software es muy amplio y variado más aun cuando se trata de transmisión de canales y digitalización de los archivos audiovisuales. Es muy común encontrarse con aplicaciones similares derivadas unas de otras, esto viene dado debido a las características específicas de cada uno de los clientes y usuarios finales. Entre los exponentes más utilizados en el ámbito del trabajo con imágenes y la transmisión de audiovisuales se pueden encontrar:

1.4.1 Videoma Broadcast Monitor (VBM).

Incorpora la grabación y monitorización multicanal de la señal de Televisión Digital Terrestres (TDT), satelital, multicast, frecuencias analógicas y CCTV². Además de ello realiza el streaming, clipping³, catalogación, búsqueda, visionado y almacenamiento del contenido en tiempo real. VBM realiza funciones básicas de gran ayuda para una posterior edición. Visionado, generación de clips para la creación de EDLs (Edit Decision Lists), inserción de marca de agua y efectos, asociación de documentos (Word, Pdf, PowerPoint, Excel) al contenido, extracción de fotogramas procedentes de un vídeo... etc. La catalogación de media se organiza de forma jerarquizada por esquemas o plantillas de catalogación, como núcleo principal del que derivan los metadatos. VBM permite la auditoría y seguimiento de mítines, spots, marketing entre otras.

VBM es una solución diseñada y dirigida principalmente a las agencias de noticias, medios de comunicación y para todas aquellas entidades que en su rutina de trabajo diario, consumen e interactúan con información procedente de televisión. (9)

1.4.2 TdTarsys.

Tarsys es una potente aplicación orientada al usuario que asegura que cualquier contenido, donde quiera que esté almacenado, pueda ser consultado por los usuarios como un único archivo. En combinación con una gestión de cintas inteligente, hace que el concepto de archivo casi en línea sea indiferente: los usuarios pueden tratar el contenido de la cinta como una parte accesible del almacén, siendo transparente para ello la ubicación de los mismos.

El contenido que está en el sistema de archivo, es indexado en la ingesta generándose una copia proxy en baja resolución. Se puede realizar cualquier trabajo sobre esta copia proxy de baja resolución

² Circuito Cerrado de Televisión.

³ Se puede traducir como: Compendio, resumen o recopilación.

Capítulo 1. Fundamentación Teórica

minimizando así las transferencias de red. De esta forma, sólo aquellas partes del archivo en alta calidad que sean necesarias serán las recuperadas, maximizando así la productividad. Soporta cualquier formato de media y encapsulado: DV, MPEG, MXF, IMX, Quick Time, WMV y H264. Posee herramientas de edición al corte en el servidor para todos los formatos de media soportados y cuenta con generación automática del storyboard. (10)

Los productos analizados anteriormente no dan solución a la necesidad de obtener o detectar logos de una media, no cumplen con los requisitos necesarios para darle solución al problema planteado. Por lo que no sería bueno usar como base alguna de ellas, aunque puedan cumplir con alguna parte del trabajo a realizar (así como la extracción de fotogramas de un video). Además poseen como principal desventaja que son implementadas sobre plataformas privadas, opción se está optando eliminar en el país. Así como el coste económico que pueda ocasionar el uso de una de estas herramientas.

1.5 Tendencias y Tecnologías para el desarrollo de software.

1.5.1 RUP. Metodología de Desarrollo.

El Proceso Unificado de Desarrollo de Software (Rational Unified Process, RUP), se destaca por tener tres características esenciales, está **dirigido por los casos de uso** donde estos representan la captura de requisitos que definen las funcionalidades del sistema. En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema, también guían su diseño, implementación y prueba. Los Casos de Uso constituyen un elemento integrador y guía. Otra de las características que posee RUP es **centrado en la arquitectura** que involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden, por último, **iterativo e incremental**, esta característica se basa en dividir el trabajo en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

El ciclo de vida de RUP divide el proceso en cuatro fases, que se especifican en la **Figura 6**, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. **Inicio**: esta fase tiene como propósito definir y acordar el alcance del proyecto, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores.

Capítulo 1. Fundamentación Teórica

Otra de las fases es la de **Elaboración**: se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar. En la fase de **Construcción**: es completar la funcionalidad del sistema, para ello se deben esclarecer los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto. Por último en la fase de **Transición**: se asegura que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas. (11)

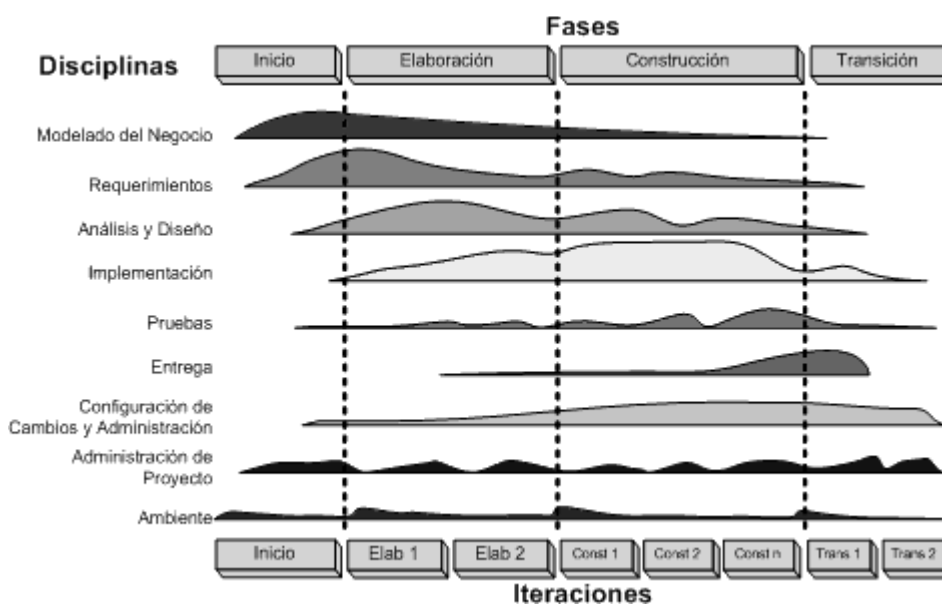


Figura 6: Fases de RUP (11)

Con el uso de la metodología RUP se tienen las siguientes ventajas: Mitigación temprana de posibles riesgos altos, progreso visible en las primeras etapas, temprana retroalimentación que se ajuste a las necesidades reales, gestión de la complejidad y conocimiento adquirido en una iteración puede aplicarse de iteración a iteración entre otras. En la **Tabla 1** se podrá observar una comparación entre la metodología RUP y Extreme Programming (XP).

Tabla 1: RUP vs XP.

Característica	RUP	XP
Tamaño del Grupo de Trabajo.	Para grupos grandes, más de 5 personas.	Para grupos pequeños, hasta 4 personas.

Capítulo 1. Fundamentación Teórica

Obtención de Requisitos.	Casos de uso.	Historia de usuarios.
Carga de Trabajo.	Mayor.	Menor.
Relación con el Cliente.	Se presentan artefactos es muy formal.	No tiene formalismo.
Documentación.	Mucha.	Poca.
Duración de Proyectos.	Largo.	Corto.
Detección de Errores.	En forma temprana.	A largo plazo.
Reutilización de Código.	Si.	Si.
Simplicidad de Diseño.	No.	Si.
Centrado en la Arquitectura.	Si.	No.
Soporte Técnico Continuo.	Menor.	Mayor.
Diseño Simple.	No.	Si.
Desarrollo.	Iterativo.	Iterativo.
Evaluación del Estado de Proyecto.	Largo.	Corto.
Accesibilidad del Código Fuente por parte del Cliente.	Poca.	Mucha.

RUP es seleccionada como metodología a utilizar debido a que no es un sistema de pasos rígidamente establecidos, sino una metodología flexible y evolutiva adaptable al contexto y necesidades de cada organización y que se compone de fases y disciplinas, lo cual permite guiar las actividades de un proyecto y contar con una documentación adecuada a lo largo del desarrollo de un proyecto. También porque provee un entorno de desarrollo confiable basado en estándares, permite que cada participante intervenga en la parte del proceso que le compete directamente, permitiendo tener claro el proceso de desarrollo a seguir.

Capítulo 1. Fundamentación Teórica

1.5.2 Visual Paradigm. Herramienta de Modelado.

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Posee una licencia gratuita y comercial, producto de calidad, soporta aplicaciones web, varios idiomas, generación de código para java y exportación como html, compatibilidad entre ediciones, fácil de instalar y actualizar. Se propone utilizar esta herramienta por las posibilidades que brinda en su uso así como la usabilidad que posee, es una herramienta fácil de utilizar y el trabajo con ella se hace cómodo para el usuario.

Por su estabilidad de ejecución en diferentes sistemas operativos y la facilidad de abrir y trabajar con un modelo UML utilizando el mismo programa sin importar el sistema operativo y sin afectar en absoluto el trabajo hecho; además destacar que esta herramienta guarda todo el modelo en un solo fichero, y basta con copiarse solo ese fichero y uno está seguro de que tiene todo el trabajo encapsulado en él.

1.5.3 C++. Lenguaje de Programación.

La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. C++ es un lenguaje de programación orientado a objetos. Se suele decir que es un lenguaje híbrido, ya que permite la programación estructurada. Permite la separación de un programa en módulos que admiten compilación independiente.

El uso de este lenguaje de programación trae consigo una serie de ventajas entre las que podemos mencionar:

- Es un lenguaje que permite programar de forma estructurada, modular y orientado a objetos. Por lo que permite la asimilación de todas estas técnicas.
- Es un lenguaje que permite el manejo de todo tipo de estructuras de datos (arrays, pilas, colas, textos, objetos,...) por lo que es capaz de resolver todo tipo de problemas.
- Es un lenguaje compilado, lo que le hace muy rápido (quizá el más rápido, descontando el ensamblador)
- Permite crear todo tipo de aplicaciones.
- Gracias a su capacidad de uso de objetos y clases, facilita la reutilización del código.
- Permite incorporar las librerías más utilizadas para hacer aplicaciones.

Capítulo 1. Fundamentación Teórica

- Es el lenguaje que sirve de base para los lenguajes modernos como Java, C#, Perl, entre otros.

Después de estudiar algunas de las características y ventajas del lenguaje de programación de C++ se elige para la creación de la herramienta ya que tiene un alto grado de optimización en memoria y velocidad, lo que lo convierte en uno de los lenguajes más eficientes y potentes entre todos.

1.5.4 Qt-Creator. Entorno de Desarrollo Integrado.

Para la realización de la herramienta se toma como Entorno de Desarrollo Integrado (IDE) Qt-Creator, ya que es un IDE multiplataforma, diseñado para hacer que el desarrollo en C++ de la aplicación Qt sea más rápido y fácil. Posee un avanzado editor de código C++. Posee también una interfaz gráfica de usuario integrada y diseñador de formularios, depurador visual, resaltado y auto-completado de código y soporte para refactorización de código.

Se propone el uso de este IDE para el desarrollo de la herramienta debido a que es un IDE de código abierto por lo cual tiene mucha aceptación en las comunidades de programación lográndose así una gran contribución por parte de estas comunidades en su desarrollo, posee un editor visual pudiendo crear interfaces. Qt-Creator combina edición, depuración, gestión de proyecto, localización y herramientas de compilación, todo lo necesario para un programador sin tener que cambiar de entornos gráficos o aplicaciones. Es un IDE multi-plataforma debido a que puede ser utilizado tanto en Windows, Linux como en Mac Os. Otra de las grandes ventajas con que cuenta Qt-Creator es el sin número de documentación que existe del mismo, desde tutoriales hasta fragmentos de código.

1.5.5 Biblioteca OpenCV.

OpenCV (Open Source Computer Vision Library) es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Se ha utilizado en infinidad de aplicaciones desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos. Open CV es multiplataforma, existiendo versiones para GNU/Linux, Mac OS X y Windows. Esta librería es desarrollada para proporcionar un entorno de desarrollo fácil de utilizar y altamente eficiente. Esto se ha logrado, realizando su programación en código C y C++ optimizados, aprovechando además las capacidades que proveen los procesadores multi núcleo.

Con el uso de esta biblioteca es posible la creación de aplicaciones muy poderosas en el dominio de visión digital. Esta biblioteca incluye muchas funciones para el procesamiento de imágenes y videos entre las que podemos mencionar: el análisis estructural, análisis de movimiento, reconstrucción 3d,

Capítulo 1. Fundamentación Teórica

segmentación y reconocimiento de objetos entre otras funciones. Por estas razones se propone utilizar OpenCV como biblioteca de apoyo para la realización de la herramienta a crear.

1.5.6 Sistema Gestor de Base de Datos

Una base de datos (BD) es un conjunto de información almacenada de forma organizada sistemáticamente para su posterior uso. Una base de datos también es considerada como un conjunto de información relacionada que se encuentra agrupada o estructurada.

Un Sistema Gestor de Base de Datos (SGBD) es una aplicación que permite crear y administrar una BD, contribuyendo a mantener una integridad, seguridad y privacidad en la misma. Un ejemplo de esto es:

1.5.6.1 PostgreSQL.

Es el sistema de gestión de bases de datos de código abierto más avanzado del mundo y en sus últimas versiones posee muchas características que solo se podían ver en productos comerciales de alto calibre. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Este SGBD tiene una serie de características que lo hacen mejor en un nivel más funcional que otros sistemas como es el aislamiento y la durabilidad. Las cuales aseguran que una operación no puede afectar a otra, lográndose así que dos transacciones sobre la misma información nunca generarán ningún tipo de error y que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema. (12)

1.5.6.2 Sqlite.

Es uno de los SGBD que más se están utilizando actualmente por su rapidez con el trabajo en base de datos pequeñas, aunque no se pueden realizar una serie de operación en la misma. Como es el uso del RIGHT JOIN ni el FULL JOIN, no permite las transacciones en cadenas, como tampoco posee ningún mecanismo de replicación y sincronización de datos. También posee en contra que cuando se realiza una operación de escritura la base de datos se bloquea, si bien se pueden definir tipos de datos, no se hace ningún tipo de validación a la hora de insertar o actualizar datos. No soporta INFORMATION_SCHEMA (usa sus propias funciones para inspeccionar al meta data). Falta de clave foránea: cuando se realice la creación de la tabla desde el modo consola, está permitiendo el uso de la clausura, aunque no realizara el chequeo de la misma.

Capítulo 1. Fundamentación Teórica

Al realizar una comparativa acerca de los SGBD expuestos anteriormente se propone el uso del Sistema Gestor PostgreSQL gracias a sus diferentes cualidades y comodidades que brinda para el trabajo con las bases de datos de grandes proporciones.

1.6 Conclusiones.

Luego de un análisis del capítulo expuesto se arriban a las siguientes conclusiones:

- Se realizó el análisis de los principales métodos y técnicas para la detección de bordes de una imagen digital.
- Se realizó un breve análisis de aplicaciones con semejanzas en el trabajo del procesamiento de imágenes digitales y detección de logos.
- Se hace mención y un breve análisis a las tendencias y tecnologías a utilizar para la realización de la herramienta.

Capítulo 2. Características del Sistema

CAPÍTULO 2. Características del Sistema

2.1 Introducción.

En este capítulo se dará a conocer la propuesta de solución para la realización de la detección de logo. También se mencionará la especificación de requisitos funcionales y no funcionales que presentará la herramienta a desarrollar, además de realizar una descripción de los casos de usos del sistema y del modelo de dominio.

2.2 Sistema propuesto.

Se propone desarrollar una herramienta que permita el procesamiento de un video para la detección y el posterior reemplazo del logo que exista en la media. Cumpliendo con el objetivo de realizar una representación clara del canal que este transmitiendo la media en ese momento. La herramienta debe ser sencilla, capaz de integrarse a una plataforma de transmisión de canales. Aportando un valor agregado a la misma permitiéndole este tipo de opciones en su trabajo.

2.3 Modelo de domino.

Durante el análisis para la implementación del sistema existe un artefacto que se genera para la especificación del negocio. El modelo de negocio se crea con el objetivo de especificar los procesos que involucra, o en que se desempeña un cliente en específico, este modelo se basa en la generación de otros dos modelos principalmente. Entre los que se encuentra el modelo del dominio, el cual es una representación visual estática de los objetos reales que intervienen en la aplicación o proyecto. Al no tener una clara visión del modelo de negocio se toma la decisión de generar el modelo de dominio. En la **Figura 7** se muestra el modelo de dominio generado.

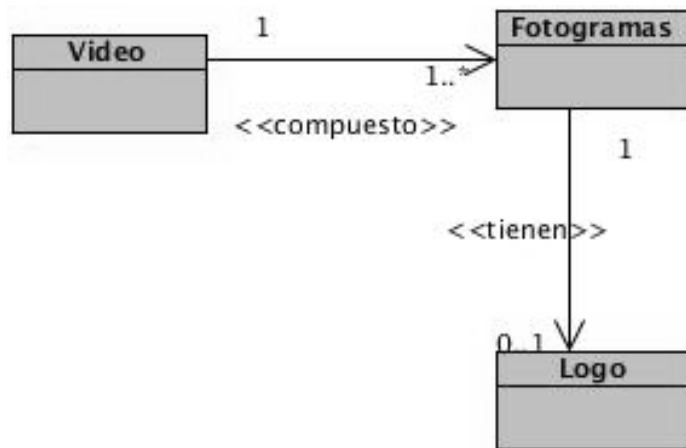


Figura 7: Diagrama del Modelo de Dominio.

Descripción del Modelo de Dominio:

El video está compuesto por uno o muchos fotogramas, estos cuentan con un logo.

Video: archivo que almacena los fotogramas de una multimedia.

Fotogramas: espacio donde se encuentra el objeto.

Logo: objeto a localizar en los fotogramas.

2.4 Especificación de requisitos de software.

2.4.1 Requisitos Funcionales.

Los requisitos funcionales no son más que una característica requerida por el sistema que denota una acción del mismo. A continuación se presentaran los requisitos funcionales de la herramienta.

RF1: Cargar una media según una dirección de almacenamiento.

RF2: Detectar la posición del logo en una media.

RF3: Reemplazar el logo de una media.

RF4: Transmitir la media con el logo reemplazado.

Capítulo 2. Características del Sistema

2.4.2 Requisitos no Funcionales.

Los requisitos no funcionales son aquellas características que presenta la herramienta que no describe funciones a realizar o información a guardar. A continuación se dará a conocer los requisitos no funcionales de la herramienta.

Usabilidad

- Establece las cualidades mínimas que debe tener el operador del ordenador donde estará la herramienta.

Eficiencia

- El tiempo de respuesta para la detección de un logo en una media de la herramienta debe ser menor de 1min.

Soporte

- La herramienta será revisada mensualmente para su mantenimiento.
- Lenguaje: se utilizará para el desarrollo de la herramienta, C++.
- IDE: Qt-Creator.
- Biblioteca: se apoyará en la biblioteca de desarrollo, OpenCV.

Documentación de usuario

- La herramienta no tendrá un manual de usuario propio ya que estará integrada con el manual del subsistema transmisión.

Interfaz

- Usuario: La herramienta contará con una interfaz sencilla y amigable.
- Hardware: La PC en la que esté la herramienta debe tener al menos 1GB de RAM, 160 GB de disco duro, un procesador Core i3 a 3.1 GHz y debe contar con una tarjeta grafica superior a los 250 MG.
- Software: Sistema Operativo: GNU/Linux (Ubuntu 11.10).

Capítulo 2. Características del Sistema

2.5 Caso de Uso del Sistema.

Un caso de uso no es más que la descripción del proceso que se llevará a cabo para la realización de una actividad. En un caso de uso pueden interactuar personas o sistemas a los cual se les denomina actores. Los diagramas de caso de uso sirven para especificar la comunicación entre el sistema y los actores.

2.5.1 Actores.

Un actor es aquella entidad externa al sistema que guarda una relación con este y que tiene una responsabilidad o funcionalidad sobre la herramienta. Esto incluye tanto a personas como sistemas externos. Véase la **Tabla 2**.

Tabla 2: Actores.

Actor	Descripción
Administrador de Transmisión	Interactuar con el sistema
Sistema	Aplicación sobre la que se trabaja

2.5.2 Diagramas de Casos de Uso.

En la **Figura 8** se muestra el Diagrama de Casos de Uso de la herramienta.

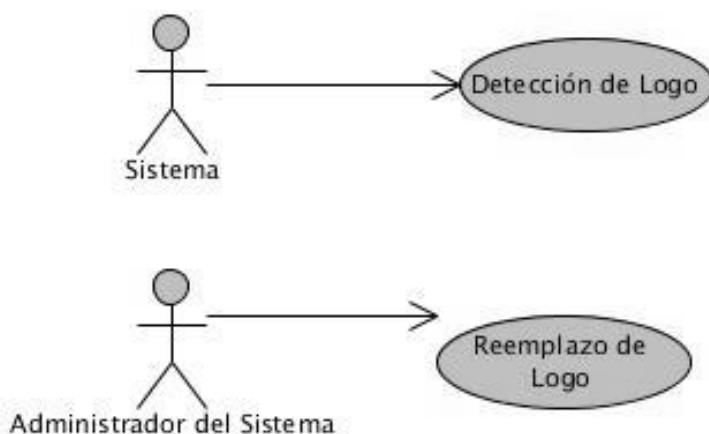


Figura 8: Diagramas de Casos de Uso.

Capítulo 2. Características del Sistema

2.5.3 Descripción de los Casos de Uso.

Tabla 3: Descripción del Caso de Uso Detección de Logo.

Caso de Uso:	Detección de Logo
Actor:	Sistema
Propósito:	Buscar el logo en la media
Resumen:	El caso de uso se inicia cuando el sistema detecta que se ha adicionado una nueva media a la base de datos.
Precondiciones:	Debe haberse insertado una media.
Referencias:	RF1, RF2
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El sistema recibe una notificación sobre la inserción de una media.	
	2. El sistema Carga el video. 3. El sistema extrae los fotogramas del video. 4. El sistema le realiza la umbralización a los fotogramas extraídos. 5. Se aplica una multiplicación lógica a los fotogramas para obtener la ubicación del logo. 6. El sistema guarda los datos del logo en una base de datos.

Capítulo 2. Características del Sistema

Tabla 4: Descripción del Caso de Uso Reemplazo de Logo.

Caso de Uso:	Reemplazo de Logo
Actor:	Administrador de Transmisión
Propósito:	Reemplazar el logo en la media
Resumen:	El caso de uso se inicia cuando el Administrador de Transmisión decide reemplazar el logo de la media que se encuentra transmitiendo.
Precondiciones:	Debe existir el logo del canal.
Referencias:	RF3, RF4
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Administrador de Transmisión marca la opción de reemplazar logo.	
	2. Reemplaza el logo por el del canal que esté transmitiendo.

2.6 Conclusiones.

En el capítulo antes descrito se abordaron diferentes particularidades del sistema propuesto, de lo que se puede concluir:

- Con la extracción de los requisitos funcionales y no funcionales, se permite conocer las funcionalidades y las restricciones que presentará la herramienta.
- Con la descripción de los Casos de Uso y del Modelo de Dominio, se puede comprender cómo interactúan los actores del sistema con la herramienta.

CAPÍTULO 3. Análisis y Diseño del Sistema

3.1 Introducción.

En este capítulo se presentará la propuesta de solución para desarrollar la herramienta. La propuesta está compuesta por los modelos de análisis y diseño, así como los diagramas de clase e interacción. Se tendrá como parte de la propuesta los patrones de diseños a aplicar para la solución. Esto servirá de base y guía para la implementación de la herramienta.

3.2 Patrón de Arquitectura.

Desde los inicios de la informática se han desarrollado formas y guías para la solución de problemas, llamándoles Arquitectura de Software, estas indican la estructura, funcionamiento e interacción entre las partes del software. La Arquitectura de Software está definida por diferentes patrones que ofrecen soluciones a problemas de la arquitectura, un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software.

Para la solución de la herramienta se ha decidido utilizar el patrón o estilo de arquitectura en Capas. Debido a que este patrón permite la separación de capas o niveles de las funcionalidades de la herramienta, o sea cada capa o nivel contiene la funcionalidad relacionada solo con las tareas de esa capa. Esta arquitectura nos ofrece un aislamiento efectivo logrando realizar actualizaciones en las capas sin afectar el resto del sistema, se puede tener un mejor rendimiento de la herramienta ya que puede mejorar la estabilidad y tolerancia a fallos. Con el uso de esta arquitectura se obtendrá una independencia de interfaces externas y las consideraciones del hardware, se tendrá una flexibilidad a la hora de la reutilización ya que la misma lo permite, así como la integración a otras aplicaciones o sistemas.

3.3 Patrones de diseño.

Los patrones de diseño dan una solución simple, elegante, probada y documentada a problemas del desarrollo de software, que ofrecen una ayuda a cómo deben quedar estructuradas las clases para el proceso de construcción del software. Entre los patrones de diseño más utilizados para la confección de un software se encuentran los patrones GRASP (General Responsibility Assignment Software Patterns) y GoF (Gang of Four).

Los patrones GoF se clasifican en patrones estructurales, creacionales y de comportamiento. Los creacionales resuelven los problemas relativos a la creación de objetos, los estructurales ofrecen

Capítulo 3. Análisis y Diseño del Sistema

soluciones a problemas relativos a la composición de los objetos. Por otra parte los de comportamiento dan solución a problemas referentes a la interacción entre objetos. (12)

Los patrones GRASP son patrones generales de software para asignación de responsabilidad, por lo que se hace más útil a la hora de la creación de la herramienta. Por lo que se propone utilizar los patrones: Experto, Creador, Controlador, Alta Cohesión y Bajo Acoplamiento.

El patrón experto es el principio básico de la asignación de responsabilidad ya que este patrón es el encargado de indicar la responsabilidad de creación o implementación que recae en cada clase que conoce la información, este patrón se ve evidenciado en cada clase de la herramienta a desarrollar. El patrón creador es el responsable de la asignación de cual clase es la responsable de la construcción de objetos, evidenciándose en todas las clases creadas. El patrón controlador, es el encargado de recibir los datos y enviarlos a los métodos implementados en las diferentes clases aumentando el control, este patrón se puede comprobar en la clase "Ubicación". El patrón de alta cohesión, este patrón hace alusión a la información de cada clase, o sea que la información que exista en cada clase debe ser coherente y relacionada con la misma, poniéndose a prueba este patrón en las clases implementadas en la herramienta. Por otra parte el patrón de bajo acoplamiento expresa la idea de mantener las clases lo menos relacionadas posible para en caso de que ocurra un cambio en una de las clases no tenga un repercusión en el resto.

Con la utilización de estos patrones se logrará tener un diseño de clases bien estructurado y flexible a los cambios de última hora que puedan ocurrir en la implementación de la herramienta, logrando así una potencial reutilización de código, control e independencia entre las clases.

3.4 Modelo de Análisis.

Para la realización de la herramienta es necesario realizar un modelo análisis, el cual está compuesto por los diagramas de Clases del Análisis y los diagramas de Interacción. Este producto de trabajo define un modelo de objeto que describe la realización de guiones de uso, y sirve como una abstracción del modelo de diseño. El modelo de análisis contiene las clases de análisis y cualquier producto de trabajo asociados. Este modelo puede ser un producto de trabajo temporal, como cuando evoluciona hacia un modelo de diseño, o puede permanecer durante algunos o todos los proyectos, y quizás más, sirviendo como visión general conceptual del sistema.

Capítulo 3. Análisis y Diseño del Sistema

3.4.1 Diagrama de Clases del Análisis.

El diagrama de clases del análisis es donde se puede encontrar las clases de tipo Interfaz, Controladora y Entidad.

- La clase interfaz se encarga de la interacción del actor con el sistema
- La clase controladora es la encargada de manejar los aspectos dinámicos del sistema, o sea coordinar las acciones y flujos de control.
- La clase entidad maneja toda la información de larga vida, es una estructura de datos lógica que contribuye a comprender la información de la cual depende el sistema.

En la **Figura 9** se muestra el diagrama de clase del análisis Detección de Logo.

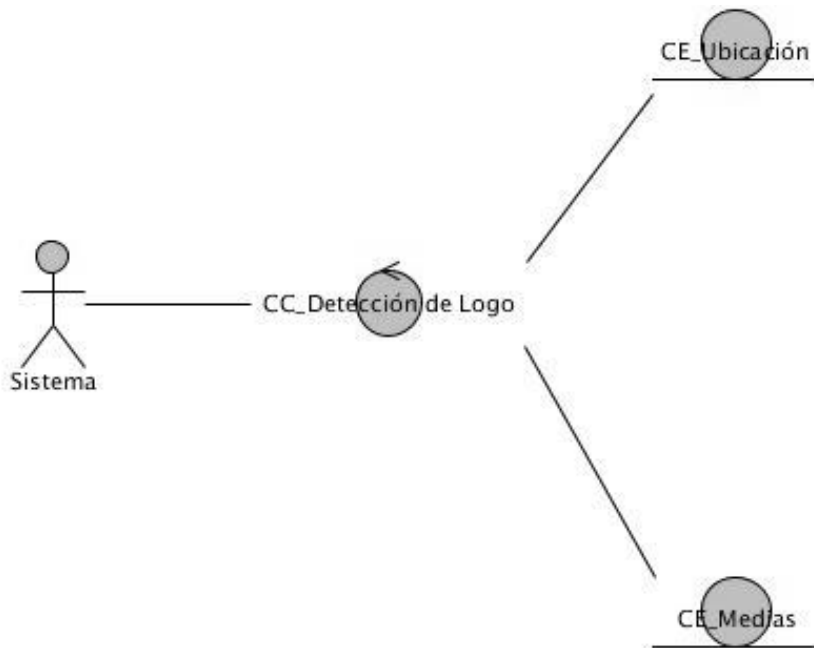


Figura 9: Diagrama de Clase del Análisis Detección de Logo.

En la **Figura 10** se podrá observar el diagrama de clase del análisis Reemplazar Logo en la cual se encuentran las clases relacionadas con esta funcionalidad.

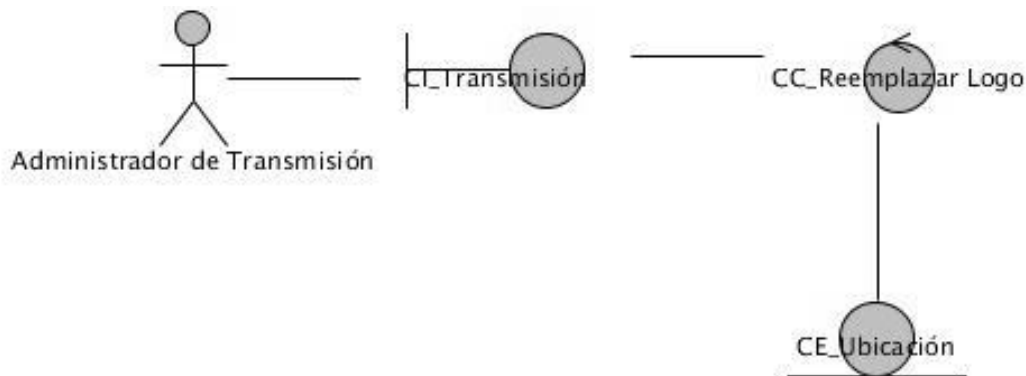


Figura 10: Diagrama de Clase del Análisis Reemplazar Logo.

3.4.2 Diagrama de Colaboración.

El diagrama de colaboración es uno de los diagramas que se pueden encontrar dentro del modelo de análisis, el diagrama de colaboración muestra la manera de interactuar las clases entre sí mediante mensajes. En las **Figuras 11 y 12** se pueden observar los diagramas de colaboración para los casos de uso Detección y Reemplazo Logo propuestos para la herramienta.

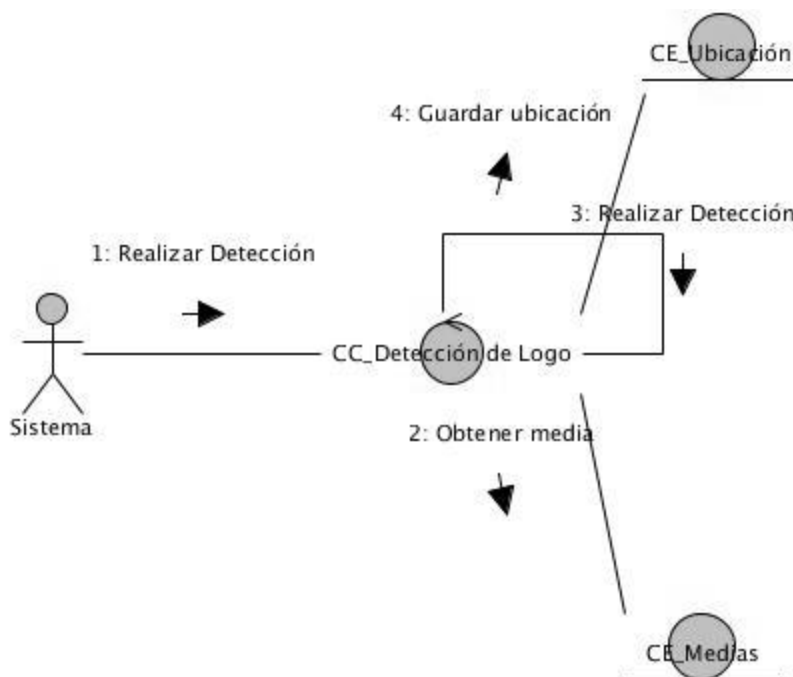


Figura 11: Diagrama de Colaboración Detección de Logo.

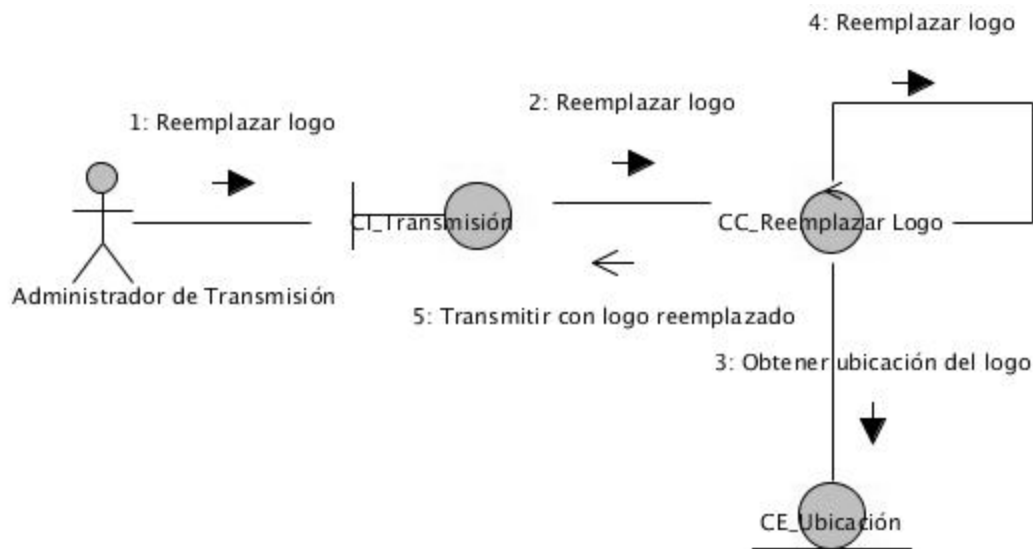


Figura 12: Diagrama de Colaboración Reemplazar Logo.

3.5 Modelo del diseño.

Este es un modelo de objeto que describe la realización de guiones de uso, y sirve como una abstracción del modelo de implementación y el código fuente. El modelo de diseño se utiliza como entrada esencial para actividades en implementación y prueba. Se utiliza para concebir y para documentar el diseño del sistema de software. Es un producto de trabajo integral y compuesto que abarca todas las clases de diseño, subsistemas, paquetes, colaboraciones y las relaciones entre ellos.

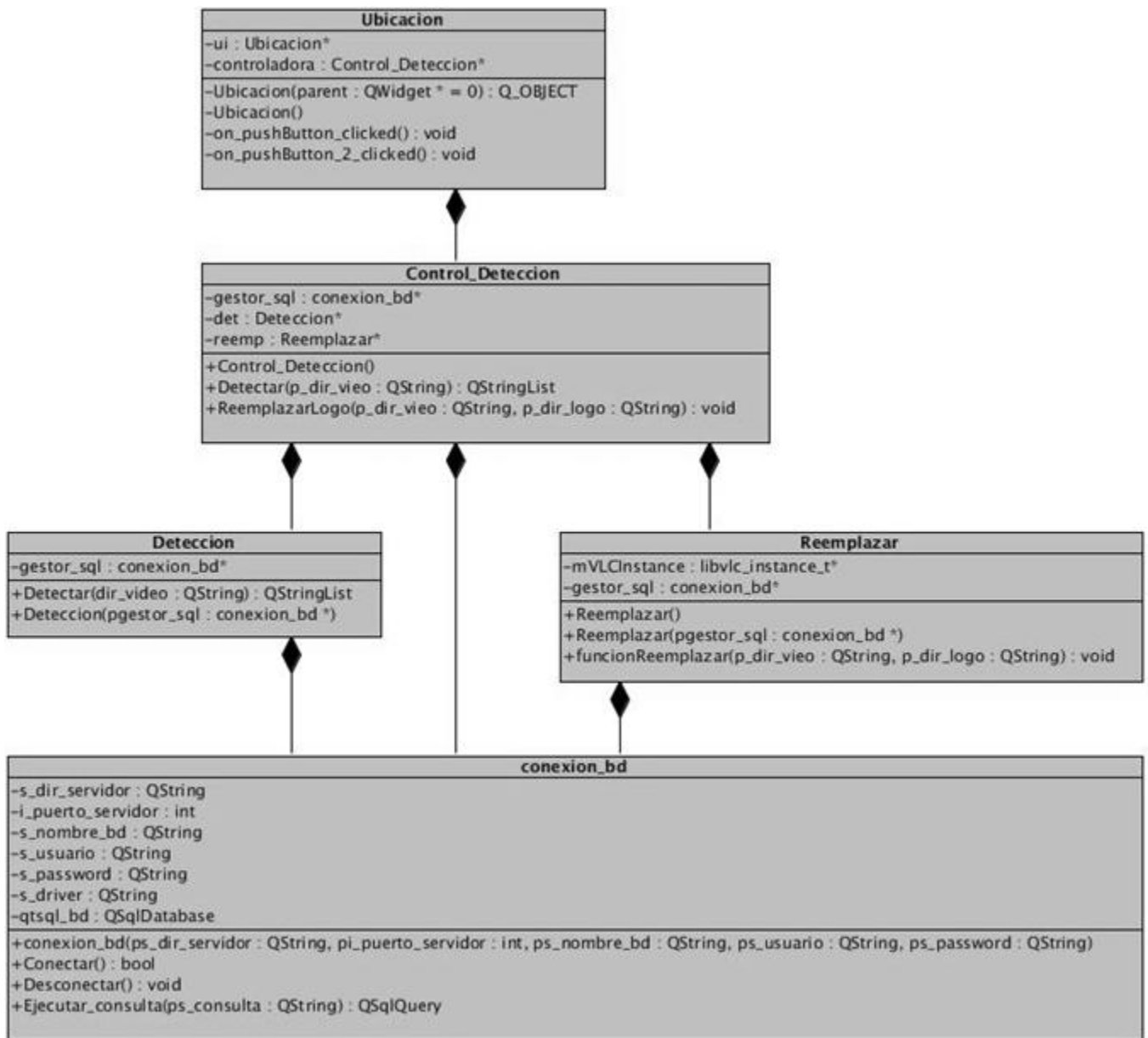


Figura 13: Diagrama de Clase del Diseño.

En la **Figura 13** se observa la distribución de las clases según su papel o funcionalidades en las diferentes capas de la herramienta. En la capa de lógica de negocio se encuentran las clases Detección y Reemplazar que por su importancia son las encargadas de controlar estas funcionalidades por separadas, para lograr una independencia de las funcionalidades entre las clases. En la capa de acceso a datos se encuentran las clases Conexión que será la encargada de crear una conexión a la base de datos para lograr obtener cualquier información de la misma. Las clases Ubicación y Media son las encargadas de brindar la información necesaria para la herramienta. Y por último la capa de

Capítulo 3. Análisis y Diseño del Sistema

Presentación donde se ubicará un botón o checkbox para invocar la funcionalidad de reemplazar cuando el usuario lo desee conveniente.

Con esta distribución se podrá contar con una total independencia entre las clases para no afectar una modificación dentro de ellas con respecto a la otra. Teniendo una limpieza y claridad a la hora de programar la herramienta.

3.6 Conclusiones.

En el capítulo se pudieron analizar diferentes aspectos de importancia para la elaboración de la herramienta propuesta como es:

- Propuesta de patrones de arquitectura y diseño para una correcta estructuración y fácil desarrollo de la herramienta, para lograr una alta flexibilidad y escalabilidad de la misma.
- Con la realización de los Modelos de análisis y diseño se puede tener una concepción más clara de la relación que existe entre las clases asociadas a la herramienta y de la estructura de las mismas.

Capítulo 4. Implementación y Prueba

CAPÍTULO 4. Implementación y Prueba

4.1 Introducción.

En el siguiente capítulo se estará abordando sobre los elementos del modelo del diseño que se implementan en términos de componentes. También se describirá el tema relacionado con el proceso de pruebas a la herramienta con el fin de garantizar el correcto funcionamiento de la misma. Se estará realizando un diseño de caso de prueba obteniendo el resultado de las mismas a la herramienta.

4.2 Estándares de Codificación.

Los estándares de codificación constituye un elemento principal en la programación, ya que por los cuales se deben regir los desarrolladores, para asegurar que el código sea legible para cualquier equipo de trabajo. Permitiendo que las líneas contengan la menor cantidad de errores posibles, a continuación se presentan algunas de las pautas a seguir en la programación utilizada en la construcción de la solución:

- Declaración de variables
 - Los nombres de las variables deben tener un relación con la acción que cumplen en la herramienta.
- Saltos de línea
 - Se añadirá un salto de línea después de un “;” (punto y coma).
- Espacios y líneas en blanco
 - Las funcionalidades deben estar separadas por dos espacios o líneas en blanco.
 - Las acciones serán separadas por un espacio o línea en blanco.
 - No utilizar espacio después de un paréntesis de apertura o de cierre.

`cvCreateImage(cvSize(img_s.width, img_s.height), IPL_DEPTH_LABEL, 1)` correcta

`cvCreateImage(cvSize(img_s.width, img_s.height), IPL_DEPTH_LABEL, 1)` incorrecta

- Utilizar espacio al realizar una operación de asignación o comparación.

`int area = anterior->area` correcta

`int area=anterior->area` incorrecta

- Concatenación
 - Para concatenar las variables se utilizara el operador “_” (under-escoart).

Capítulo 4. Implementación y Prueba

4.3 Análisis del algoritmo

En el procesamiento digital de imágenes es de suma importancia identificar los objetos de interés en las imágenes, esto puede ser posible mediante el uso de los blobs, estos no son más que las regiones (u objetos) de mucha o poca relevancia en una imagen digital. A continuación se presentaran los pasos y técnicas utilizadas para lograr el correcto procesamiento de una imagen digital para lograr una detección de logo satisfactoriamente.

El procesamiento se inicia con la captura de un video (o media) mediante una dirección específica que será donde se encuentre almacenado el mismo y se verifica que la captura haya sido satisfactoria. Se procede a la extracción de los fotogramas asociados al video (o media) aplicándoles los algoritmos siguientes:

- **Conversión de la imagen digital**

La conversión de una imagen digital consiste en llevar la imagen de un color a otro. Es necesario llevar la imagen de tres canales a una de un solo canal, o sea de una imagen a color a una en escala de grises mediante la siguiente función:

$$RGB \text{ to Gray: } \leftarrow 0.299 * R + 0.587 * G + 0.114 * B$$

Donde R , G , B son los valores de un pixel de la imagen de tres canales en rojo, verde y azul respectivamente y Y el valor que tomara ese pixel en la imagen de un solo canal. En la **Figura 14** se muestra un ejemplo de la conversión de una imagen digital.



Figura 14: Conversión de imagen.

Capítulo 4. Implementación y Prueba

- **Binarización de la imagen**

Esta técnica se le aplica a una imagen de un solo canal, a la cual se le ha fijado un umbral. Se utiliza normalmente para obtener un doble nivel (binario) de la imagen en escala de grises o para retirar ruidos, es decir, el filtrado de píxeles con valores demasiado pequeño o demasiado grande mediante la siguiente forma:

$$dst(x,y) = \begin{cases} maxVal & \text{if } src(x,y) > thresh \\ 0 & \end{cases}$$

Donde *maxVal* sería el valor máximo que se le otorgaría a un píxel de la imagen (255), *src(x,y)* es el valor del píxel de la imagen en escala de grises, *thresh* es el umbral fijado en la imagen y *dst(x,y)*, el valor que tomara el píxel de la imagen resultante. Esto puede ser visible en la **Figura 15**.



Figura 15: Binarización de imagen.

- **Multiplicación lógica**

El proceso de multiplicación entra en vigor luego de tomarse dos imágenes o fotogramas como muestra y haberles realizado los procedimientos de conversión y binarización correspondientes a cada imagen. Estas dos imágenes entran en un ciclo de comparación entre sí, chequeándose el valor de los píxeles de las mismas. En caso de que los valores de estos píxeles sean diferentes a la primera imagen tomada se le asignará el valor 0, y se repetirá este proceso para una *x* cantidad de muestras de imágenes siguientes. Logrando obtener así una imagen clara del contorno del logo. En la **Figura 16 y 17** se observa con claridad el proceso antes descrito.



Figura 16: Multiplicación lógica.



Figura 17: Logo del video.

- **Posición del logo**

Después de haber realizado los procesos correspondientes a la conversión, binarización y multiplicación lógica se procede a encontrar que posición o lugar tiene el logo en el video. Para esto es necesaria la ayuda de la biblioteca *cvBlob*, la cual utiliza algoritmos de componentes conexas para la obtención de objetos y algunos descriptores como área, perímetros y rectángulo envolvente (Bounding Box). Este último utilizado para encapsular el logo del video y obtener mediante su tamaño la posición del mismo. A continuación en la **Figura 18** se muestra la utilización de la biblioteca *cvBlob* en el proceso de obtención de la posición del logo.



Figura 18: Imagen con rectángulo envolvente

4.4 Modelo de Implementación.

El modelo de implementación describe como los elementos del Modelo de Diseño se implementan en términos de componentes físicos, para que puedan comprenderse y gestionarse mejor. Describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación. Como también es utilizado para describir el o los lenguajes de programación utilizados y como dependen los componentes unos de otros.

4.4.1 Diagrama de Componente.

En la **Figura 19** se observa el diagrama de componentes en el cual se muestra como está dividido el sistema en componentes y las dependencias entre ellos. Así como provee una vista arquitectónica del sistema, ayuda a los desarrolladores a visualizar caminos de implementación y a la toma de decisiones respecto a las tareas a implementar. Es utilizado para al modelación de archivos, códigos fuentes, librerías y ejecutables.

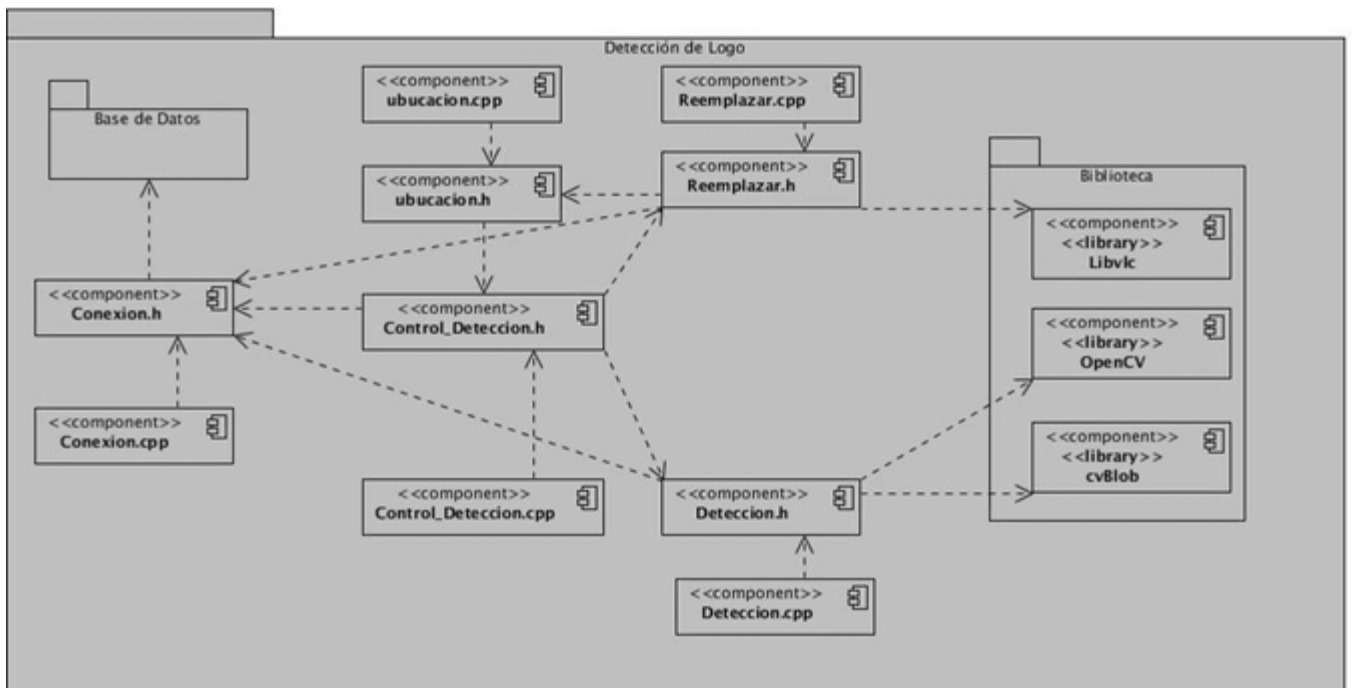


Figura 19: Diagrama de Componentes.

4.4.2 Diagrama de Despliegue.

El diagrama de despliegue permite modelar la disposición física o la tipología de un sistema, así como las conexiones físicas entre el software y el hardware.



Figura 20: Diagrama de Despliegue.

En la **Figura 20** se muestra el diagrama de despliegue propuesto para la herramienta desarrollada representado por 2 nodos. El nodo PC-Sistema donde se encontrará ubicada la herramienta y el nodo servidor de Base de Datos comunicándose entre sí mediante el protocolo TCP/IP.

4.5 Pruebas de Software.

En este proceso se ejecutan pruebas dirigidas a componentes del software o al software en su totalidad. Con el objetivo de medir la eficiencia y calidad que tiene el software para cumplir sus funcionalidades o requerimientos. Es necesario probar el software para descubrir (y corregir) la mayor

Capítulo 4. Implementación y Prueba

cantidad de errores posibles, diseñando una serie de casos de prueba que tengan una alta probabilidad de encontrar errores.

4.5.1 Plan de Prueba.

Un plan de prueba integra diseños de casos de pruebas de software en una serie de pasos que dará a lugar a una eficaz construcción de software. Un plan de prueba proporciona una descripción de pasos a seguir como parte del mismo, indica cuando se planean y cuando se dan estos pasos. Además de cuanto esfuerzo, tiempo y recursos se consumirá. Por tanto cualquier plan de prueba debe incorporar el diseño de casos de prueba, la ejecución, recolección y evaluación de los datos resultantes en la pruebas.

4.5.2 Prueba de Caja Blanca.

Se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que examinen que están correctas todas las condiciones y/o bucles para determinar si el estado real coincide con el esperado o afirmado. Esto genera gran cantidad de caminos posibles por lo que hay que dedicar esfuerzos a la determinación de las condiciones de prueba que se van a verificar.

La prueba de caja blanca se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

1. Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
2. Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
3. Ejecuten todos los bucles en sus límites operacionales.
4. Ejerciten las estructuras internas de datos para asegurar su validez.

Es por ello que se considera a la prueba de Caja Blanca como uno de los tipos de pruebas más importantes que se le aplican a los software, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.

Capítulo 4. Implementación y Prueba

Prueba del Camino Básico.

Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el Grafo de Flujo asociado y se calcula su complejidad ciclomática. Los pasos que se siguen para aplicar esta técnica son:

A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.

1. Se calcula la complejidad ciclomática del grafo.
2. Se determina un conjunto básico de caminos independientes.
3. Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

4.5.3 Diseño de casos de prueba.

CU: Detección de Logo.

Código Fuente: Detección.

Paso 1: Grafo de Flujo asociado.

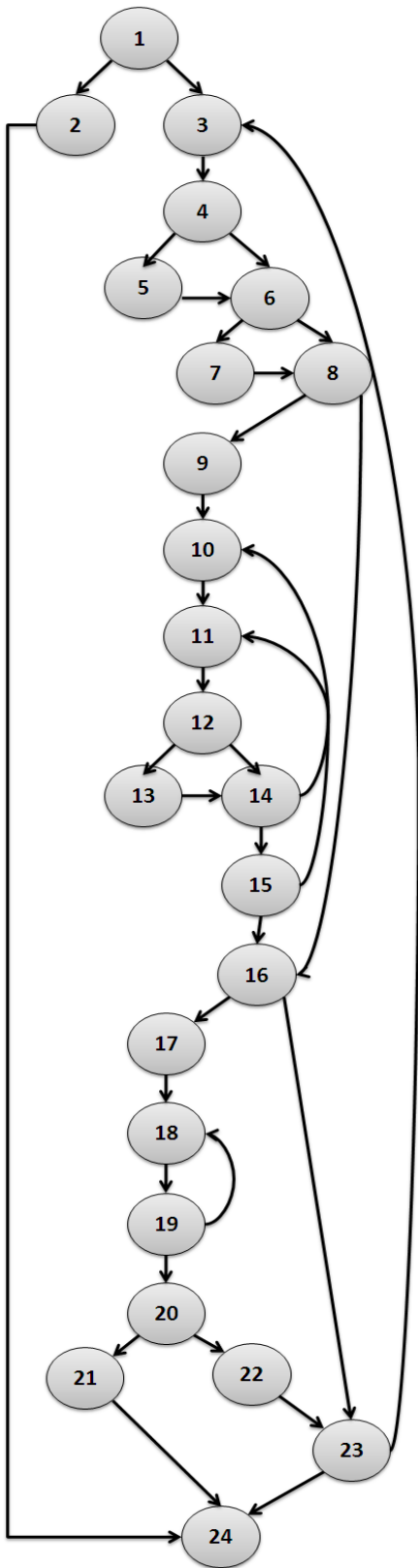


Figura 21: Grafo de Flujo.

Capítulo 4. Implementación y Prueba

Paso 2: Cálculo de la Complejidad Ciclomática del Grafo.

$V(G) = A - N + 2$, donde A es el número de aristas y N el número de nodos.

$$V(G) = 34 - 24 + 2, V(G) = 12$$

Paso 3: Conjunto básico de caminos independientes (CI).

CI.1: 1-2-24

CI.2: 1-3-4-5-6-7-8-9-10-11-12-13-14-11-12-13-14-15-10-11-12-13-14-11-12-13-14-15-16-17-18-19-18-19-20-21-24

CI.3: 1-3-4-6-8-16-23-24

CI.4: 1-3-4-5-6-7-8-9-10-11-12-13-14-11-12-13-14-15-10-11-12-13-14-11-12-13-14-15-16-17-18-19-18-19-20-22-23-24

CI.5: 1-3-4-6-8-9-10-11-12-14-15-16-23-24

CI.6: 1-3-4-5-6-8-16-17-18-19-20-21-18-19-20-21-24

CI.7: 1-3-4-5-6-7-8-9-10-11-12-13-14-15-16-23-24

CI.8: 1-3-4-5-6-7-8-16-24

CI.9: 1-3-4-5-6-7-8-9-10-11-12-14-15-16-17-18-19-20-21-24

CI.10: 1-3-4-5-6-7-8-9-10-11-12-14-15-16-23-24

CI.11: 1-3-4-5-6-8-16-24

CI.12: 1-3-4-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-24

Paso 4: Caso de Prueba para el CU Detección de Logo.

CI.1: 1-2-24

Entrada: media.

Resultado Esperado: Detectar posición del logo.

Capítulo 4. Implementación y Prueba

CI.2: 1-3-4-5-6-7-8-9-10-11-12-13-14-11-12-13-14-15-10-11-12-13-14-11-12-13-14-15-16-17-18-19-18-19-20-21-24

Entrada: media.

Resultado Esperado: Detectar posición del logo.

CI.3: 1-3-4-6-8-16-23-24

Entrada: media.

Resultado Esperado: Detectar posición del logo.

CI.4: 1-3-4-5-6-7-8-9-10-11-12-13-14-11-12-13-14-15-10-11-12-13-14-11-12-13-14-15-16-17-18-19-18-19-20-22-23-24

Entrada: media.

Resultado Esperado: Detectar posición del logo.

Tabla 5: Resultado de los Casos de Prueba.

No del camino	Resultado Esperado	Resultado de la Prueba
CI.1	Detectar posición del logo (media no capturada).	Satisfactorio
CI.2	Detectar posición del logo.	Satisfactorio
CI.3	Detectar posición del logo (contador y cantidad de muestra mal definidos).	Satisfactorio
CI.4	Detectar posición del logo (logo no encontrado).	Satisfactorio

4.5.4 Prueba de Eficiencia

Uno de los principales métodos para la verificación de que tan bueno es el software, es la utilización de este tipo de prueba, el cual mediante una serie de cálculos según la respuesta del software y la observación precisa de la persona se logra determinar los márgenes de errores que contiene el mismo. La *precisión* (P) y el *recall* (R) han sido utilizados regularmente para la medición del rendimiento y la

Capítulo 4. Implementación y Prueba

eficiencia de un software. *Precisión* trata los errores de sustitución y de inserción mientras que *recall* los de sustitución y eliminación, debido a que se necesita obtener una media única de rendimiento es que tienen tres tipos de errores: las sustituciones, eliminaciones e inserciones. Por lo que se ha determinado la media de rendimiento a través de (F) que es definida por la combinación del *recall* y la *precisión*.

Para cada prueba que se determine se supone que exista una *referencia*, que no es más que la representación de la realidad o la verdad de lo que se tiene. Cada sistema o software que participe en la prueba produce una respuesta o *hipótesis*, la prueba es utilizada para alinear la *hipótesis* con la *referencia* verificando las ranuras correctas o no. En caso de no ser correctas el error será determinado en errores de sustitución (ranura incorrecta), eliminación (ranura falsa) o inserción (ranura falsa). La puntuación de estas ranuras será utilizada para el cálculo de la media del rendimiento del software, para un mejor entendimiento de las pruebas que se realizan a continuación se tiene que:

N = número total de ranuras en la *referencia*

M = número total de ranuras en la *hipótesis*

C = número de ranuras correctas - las ranuras de la *hipótesis* de que se alineen con las ranuras de la *referencia* y se califican como correcta

S = número de sustituciones (ranuras incorrectas) - ranuras en la *hipótesis* de que se alineen con las ranuras de la *referencia* y se calificará como incorrecta

D = número de supresiones (ranuras que faltan o falsos negativos) - ranuras en la *referencia* que no se alinean con las ranuras en la *hipótesis*

I = número de inserciones (ranuras falsas o falsos positivos) - ranuras en la *hipótesis* de que no se alinean con las ranuras de la *referencia*

Definiéndose a partir de lo anterior expresado lo siguiente:

$$N = C + S + D$$

$$M = C + S + I$$

El número total de sustituciones correctas y ranuras eliminadas es equivalente al número total de ranuras en la *referencia* (N), mientras que a su vez el valor de la *hipótesis* (M) puede variar para cada

Capítulo 4. Implementación y Prueba

sistema o software que se pruebe, M puede ser mayor que N dependiendo de que si las inserciones son mayores que las eliminaciones.

La *precisión* y el *recall* son definidos por:

$$P = \frac{C}{M} = \frac{C}{C + S + I}$$

$$R = \frac{C}{N} = \frac{C}{C + S + D}$$

La *precisión* es el porcentaje de ranuras en la *hipótesis* de que son correctas, mientras que el *recall* es el porcentaje de ranuras de *referencia* para que la *hipótesis* sea correcta. La *precisión* tiene en cuenta los errores de sustitución y de inserción, mientras que el *recall* tiene en cuenta los errores de sustitución y eliminación. Definiéndose F de la siguiente manera:

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

Sabiendo que F es la media de rendimiento o eficiencia cuanto mayor sea su valor mejor tenemos en cuenta el rendimiento del sistema. Podemos entonces definir $E = 1 - F$ como la "medida de error" correspondiente.

Para la evaluación del método de detección se procesaron varios videos, como parte de la muestra se presenta detalladamente el resultado de la prueba. (14)

PRUEBAS EXPERIMENTALES

Tabla 6: Resultado del video Evolution

Parámetro	Valor
Referencia (N)	1
Hipótesis (M)	1
Número de ranuras correctas (C)	1
Número de sustituciones (S)	0
Número de supresiones (D)	0
Número de inserciones (I)	0
Precisión (P)	1
Recall (R)	1
Media de rendimiento (F)	100 %
Media de error (E)	0 %
Falsos positivos (Fp)	0
Falsos negativos (Fn)	0

En la **Figura 22** se muestran imágenes representando la respuesta del sistema o software.

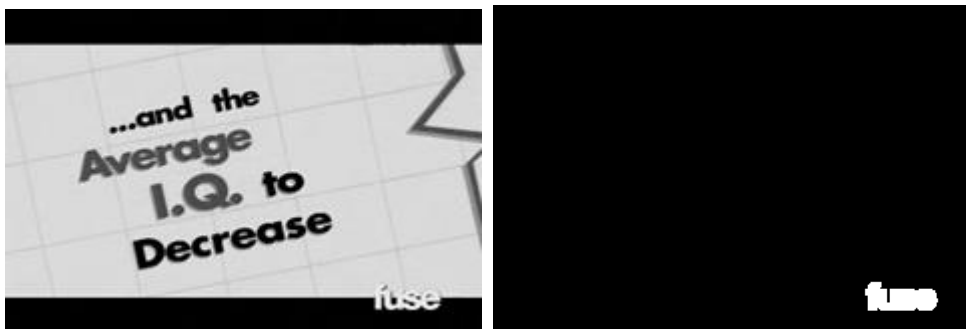


Figura 22: Imagen del video Evolution.

Tabla 7: Resultado del video Grimm.

Parámetro	Valor
Referencia (N)	1
Hipótesis (M)	0
Número de ranuras correctas (C)	0
Número de sustituciones (S)	0
Número de supresiones (D)	1
Número de inserciones (I)	0
Precisión (P)	0
Recall (R)	1
Media de rendimiento (F)	0 %
Media de error (E)	100 %
Falsos positivos (Fp)	0
Falsos negativos (Fn)	1

En la **Figura 23** se muestran imágenes representando la respuesta del sistema o software.

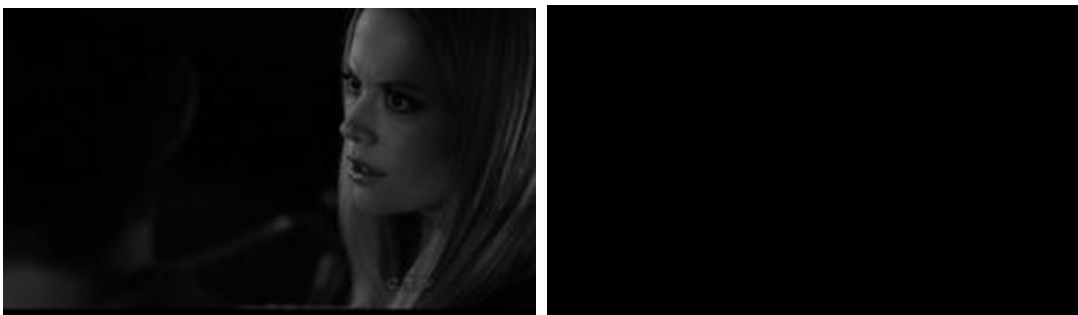


Figura 23: Imágenes del video Grimm.

Tabla 8: Resultado del video Sky.

Parámetro	Valor
Referencia (N)	1
Hipótesis (M)	2
Número de ranuras correctas (C)	1
Número de sustituciones (S)	0
Número de supresiones (D)	0
Número de inserciones (I)	1
Precisión (P)	0.5
Recall (R)	1
Media de rendimiento (F)	60 %
Media de error (E)	40 %
Falsos positivos (Fp)	1
Falsos negativos (Fn)	0

En la **Figura 24** se muestran imágenes representando la respuesta del sistema o software.



Figura 24: Imágenes del video Sky.

Capítulo 4. Implementación y Prueba

Tabla 9: Resultados del video Day.

Parámetro	Valor
Referencia (N)	2
Hipótesis (M)	2
Número de ranuras correctas (C)	2
Número de sustituciones (S)	0
Número de supresiones (D)	0
Número de inserciones (I)	0
Precisión (P)	1
Recall (R)	1
Media de rendimiento (F)	100 %
Media de error (E)	0 %
Falsos positivos (Fp)	0
Falsos negativos (Fn)	0

En la **Figura 25** se muestran imágenes representando la respuesta del sistema o software.

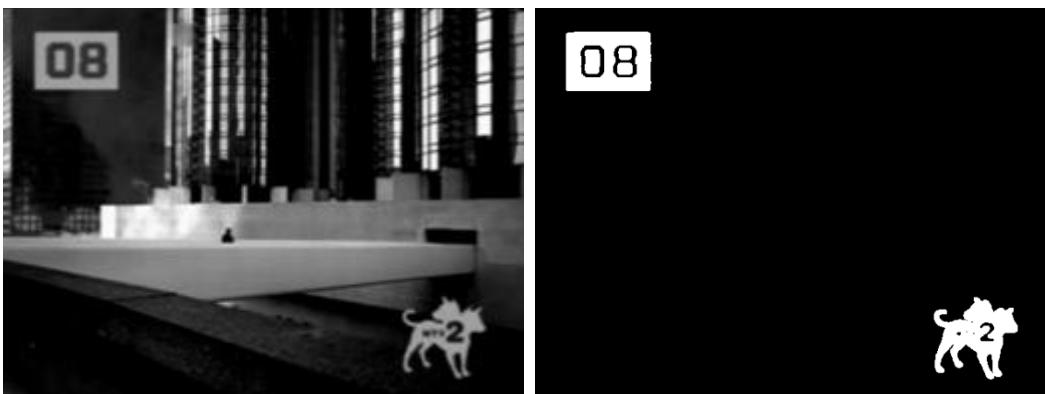


Figura 25: Imágenes del video Day.

Capítulo 4. Implementación y Prueba

En la **Tabla 10** se muestran los resultados obtenidos tras hacer la prueba a once videos de diferentes formatos y con logos en diferentes posiciones.

Tabla 10: Resultado de prueba (once videos).

Parámetro	Valor
Referencia (N)	12
Hipótesis (M)	12
Número de ranuras correctas (C)	11
Número de sustituciones (S)	0
Número de supresiones (D)	1
Número de inserciones (I)	1
Precisión (P)	9.5
Recall (R)	11
Media de rendimiento (F)	96 %
Media de error (E)	4 %
Falsos positivos (Fp)	1
Falsos negativos (Fn)	1

Para la muestra de once videos a la que se le aplico la prueba de eficiencia arrojando resultados como la existencia de un 96 % de efectividad o rendimiento en el software y de solo un 4 % en la media de errores, con solo un falso positivo y un falso negativo. Comportándose satisfactoriamente durante el transcurso del proceso de prueba se puede concluir que el software se encuentra totalmente funcional.

4.6 Conclusiones.

En el capítulo antes expuesto se pudo tener:

- Mejor visión de la distribución física y lógica de la herramienta mediante la realización de los diagramas de componentes y de despliegue.
- Una descripción clara y detallada sobre el algoritmo utilizado para la detección de logos en una media.
- Con las pruebas de caja blanca se demostró que la herramienta cumple con los requisitos necesarios, por lo que se concluye que el resultado fue satisfactorio.
- Con el uso de las pruebas de eficiencia se comprobó que por ciento de efectividad tiene la herramienta para realizar su trabajo arrojando resultados satisfactorios como un 96 % de efectividad y una media de error de una 4 %.

Conclusiones Generales

CONCLUSIONES GENERALES

Con la culminación y los resultados obtenidos en el presente trabajo de investigación se llegó a las siguientes conclusiones:

- Con el estudio de las técnicas y métodos existentes el procesamiento de imágenes digitales se determinó cuáles eran las óptimas para la detección de logos, estableciendo las bases teóricas y tecnológicas para el desarrollo de la herramienta.
- Se obtuvo la documentación necesaria que sirve de base para el desarrollo de herramientas similares o para posteriores iteraciones de la solución propuesta.
- Se describió el algoritmo utilizado para la detección de logos en una media, logrando tener una idea clara y concisa sobre los pasos, técnicas y métodos utilizado para el desarrollo de la herramienta.
- Con la validación de la herramienta mediante el diseño y aplicación de las pruebas de Caja Blanca se pudo verificar la calidad de la solución propuesta. Los resultados arrojados durante las pruebas fueron positivos indicando el cumplimiento de los requisitos funcionales.
- Con la utilización de las pruebas de eficiencia se comprobó que la herramienta posee un 96 % de efectividad o rendimiento ante su trabajo de detectar logos en una media y un 4 % de media como error.

Por lo anteriormente expuesto se concluye que los objetivos trazados para el presente trabajo han sido cumplidos satisfactoriamente. La herramienta está lista para su utilización y explotación con la capacidad de detectar y transmitir un logo deseado.

RECOMENDACIONES

Después de realizada la investigación se recomienda:

- Continuar el desarrollo de la herramienta utilizando otras técnicas de detección para enriquecer y ampliar la funcionalidad de la misma, como son las técnicas para la detección de logos con transparencias o en movimiento.
- Mejorar las técnicas de detección con el fin de reducir el tiempo de ejecución.

BIBLIOGRAFÍA REFERENCIADA

1. **Gonzalez, Rafael C. y Woods, Richard E.** *Digital Image Processing*. 2002.
2. Definición ABC. [En línea] 2011. <http://definicionabc.com>.
3. Mastermagazine. [En línea] 2011. <http://mastermagazine.com>.
4. Definición. de. [En línea] 2011. <http://definicion.de/logo>.
5. **Sáez Peña, Edmundo.** *Segmentación Automática de Video*. 2006.
6. **Jinqiao, Wang, Lingyu, Duan y Zhenglong, Li.** *A Robust Method for TV logo tracking in video streams*. 2006.
7. Detección de bordes en una imagen. [En línea] 2005. www4.ujaen.es/~satorres/practicass/practica3_vc.pdf.
8. **Valverde Rebaza, Jorge.** *Detección de bordes mediante el algoritmo de Canny*. 2011.
9. Soluciones Videoma Broadcast Monitor. *ISID*. [En línea] 2009. http://www.isid.es/spanish/product/producto_v01.htm.
10. Tedral . [En línea] 2010. <http://www.tedral.com/en/products/tarsys/product-overview>.
11. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. Madrid : s.n., 2000.
12. PostgreSQL Cuba. [En línea] <http://postgresql.uci.cu/node/63>.
13. **Cooper, James W.** *The Design Patterns Java Companion*. 1998.
14. **Makhoul, John, y otros.** *PERFORMANCE MEASURES FOR INFORMATION EXTRACTION*.

BIBLIOGRAFÍA

1. **Gonzalez, Rafael C. y Woods, Richard E.** *Digital Image Processing*. 2002.
2. Definición ABC. [En línea] 2011. <http://definicionabc.com>.
3. Mastermagazine. [En línea] 2011. <http://mastermagazine.com>.
4. Definición. de. [En línea] 2011. <http://definicion.de/logo>.
5. **Sáez Peña, Edmundo.** *Segmentación Automática de Video*. 2006.
6. **Jinqiao, Wang, Lingyu, Duan y Zhenglong, Li.** *A Robust Method for TV logo tracking in video streams*. 2006.
7. Detección de bordes en una imagen. [En línea] 2005. www4.ujaen.es/~satorres/practicas/practica3_vc.pdf.
8. **Valverde Rebaza, Jorge.** *Detección de bordes mediante el algoritmo de Canny*. 2011.
9. Soluciones Videoma Broadcast Monitor. *ISID*. [En línea] 2009. http://www.isid.es/spanish/product/producto_v01.htm.
10. Tedral . [En línea] 2010. <http://www.tedral.com/en/products/tarsys/product-overview>.
11. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. Madrid : s.n., 2000.
12. PostgreSQL Cuba. [En línea] <http://postgresql.uci.cu/node/63>.
13. **Cooper, James W.** *The Design Patterns Java Companion*. 1998.
14. **Makhoul, John, y otros.** *PERFORMANCE MEASURES FOR INFORMATION EXTRACTION*.
15. **Meisinger, Katrin, Troeger, Tobias y Zeller, Marcus.** *AUTOMATIC TV LOGO REMOVAL USING STATISTICAL BASED LOGO*. s.l. : University of Erlangen-Nuremberg.
16. **Molina, R.** *Introducción al procesamieto y Análisis de Imágenes Digitales*. 1998.
17. **Gonzalez, Rafael C. y Woods, Richard E.** *Digital Image Processing*. s.l. : Prentice Hall, 2002.
18. **Acharya, Tinku y Ray, Ajoy K.** *Image Processing. Principles and Applications*. 2005.

19. **Agam, Gady.** *Introduction to programming with OpenCV.* 2006.
20. *OpenCV: Image Processing and Computer Vision Reference Manual.* 2006.
21. **Gonzales Marcos, Ana, Alba Elias, Fernando y Castejón limas, Manuel.** *Técnicas y algoritmos básicos de división artificial.* 2006.
22. **Bradski, Gary y Kaehler, Adrian.** *Learning OpenCV.* 2008.
23. **Pham, Anh, Delalandre, Mathieu y Barrat, Sabine.** *A contour-based method for logo detection.* 2011.
24. **Li, Wenju y Li, Ling.** *A Novel Approach for Vehicle-logo Location Based on Edge Detection and Morphological Filter.* 2009.
25. **Meisinger, Katrin y Troeger, Tobias.** *AUTOMATIC TV LOGO REMOVAL USING STATISTICAL BASED LOGO DETECTION AND FREQUENCY SELECTIVE INPAINTING.*
26. **Alba, José Luis, Cid, Jesús y Mora, Inmaculada.** *Métodos de análisis de imágenes.* 2006.
27. **Zhu, Guangyu y Doermann, David.** *Automatic Document Logo Detection.*
28. **Satterwhite, Brandon y Marques, Oge.** *Automatic Detection of TV Commercial.* 2004.
29. **Wang, Hongye.** *Document Logo Detection and Recognition Using Bayesian Model.* 2010.
30. **Reynoso, Carlos y Kicillof, Nicolás.** *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft.* 2004.
31. **Li, Zhe, Schulte-Austum, Matthias y Neschen, Martin.** *Fast Logo Detection and Recognition in Document Images.* 2010.
32. **Wang, Jinqiao, y otros.** *Automatic TV Logo Detection, Tracking and Removal in Broadcast Video.* 2007.
33. **De la Torre Llorente, Cesar, y otros.** *Guía de Arquitectura N-Capas Orientada al Dominio.* 2010.
34. **Alberto Albiol, María Jose Ch. Fullá, Antonio Albiol.** *DETECTION OF TV COMMERCIALS.* 2010.
35. **Carlos Díaz, Luis, Carri, Angela y Alvarado, Deicy.** *Análisis y Diseño.* 2008.

Bibliografía

36. **Perry, Wolf.** *LA ARQUITECTURA DE SOFTWARE. EL MODELO 4+1.* 1992.
37. **Wang, Hongye y Chen, Youbin.** *Logo Detection in Document Images Based on Boundary Extension of Feature Rectangles.* 2009.
38. **Zhu, Guangyu y Doermann, David.** *Logo Matching for Document Image Retrieval.* 2009.
39. **Miralles Pechuán, Luis.** *Uso de ontologías para guiar el proceso de segmentación de imágenes.* 2007.
40. *Pressman.* 2002.
41. **Laganière, Robert.** *OpenCV 2 Computer Vision Application Programming Cookbook.* 2011.
42. *OpenCV Reference Manual.* 2010.
43. *OpenCV 2.1 Cheat Sheet (C++).*
44. **Aleshin, Denis.** *OpenCV Tutorial.*
45. **Hayet, J.B.** *Introducción al uso de OpenCV.* 2007.
46. **Kuangrong, HAO, Yunjie, WANG y Yongsheng, DING.** *Recognition of Robot Logo Based on Haar-like Features.* 2010.
47. **Dang, Zhonghua, y otros.** *Replay Detection Based on Semi-automatic Logo Template.* 2007.
48. **Zhang, Shuai, Zhang, Ze y Wang, Hai-tao.** *Research Of Robot Color Logo Orientation Based On Hough Transform.* 2010.
49. **FLORES, MIRIAN MILAGROS DÍAZ.** *PRACTICANTE DE TOO .*
50. *Segmentación por Umbralización.* 2005.
51. **Igual, Raúl y Medrano, Carlos .** *Tutorial de OpenCV.* 2008.
52. **Mateos, Ginés García.** *PROCESAMIENTO AUDIOVISUAL.*
53. **AmericaTI.com.** *Ventajas y Desventajas: Comparación de los Lenguajes C, C++ y Java.* 2006.
54. **Yan, Wei-Qi, Wang, Jun y Kankanhall, Mohan S.** *Automatic video logo detection and removal.* 2005.

Bibliografía

55. .N, Venkatesh, B, Rajeev y M., Girish Chandra. *Novel TV Commercial Detection in Cookery Program Videos*. 2009.

GLOSARIO DE TÉRMINOS

Binarizar: Llevar una imagen a escala de grises o sea blanco y negro.

CIF: (Formato intermedio común), también conocido como FCIF (Formato intermedio común completo), es un formato usado para estandarizar el horizontal y vertical resoluciones en pixeles de YCbCr secuencias en las señales video.

Convolución: La convolución de f y g se denota $f * g$. Se define como la integral del producto de ambas funciones después de desplazar una de ellas una distancia η .

Discretización: Es el proceso mediante el cual los valores se incluyen en depósitos para que haya un número limitado de estados posibles. Los depósitos se tratan como si fueran valores ordenados y discretos. Dependerá del número de muestreo dado en Hz, y esto quiere decir que si tienes una señal conocida digamos una senoidal, para poderla pasar a digital, es necesario que tomes pequeñas muestras de esa señal, después la codificas, cuantificas y tienes tu señal en forma digital.

Grafico rasterizado: Es una estructura o fichero de datos que representa una rejilla Rectangular de Píxeles o puntos de color, denominada raster, que se puede visualizar en un Monitor de Ordenador, papel u otro dispositivo de representación.

Muestreo equiespaciado: Se entiende por muestreo equiespaciado que las muestras sean puntos equidistantes en el plano.

Spot: Película de muy corta duración, generalmente de carácter publicitario.

Umbralización: También conocida como thresholding consiste en eliminar los valores superiores o inferiores (poniéndolos a cero) respecto a un valor conocido como umbral.