



**Facultad 6**

*Evaluación de la calidad de los videos en la plataforma  
PTARTV basado en métricas de distorsión estructural.*

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

**Autores:**

Pedro Diosmel Salgado Magaña

Yelen Díaz Saez

**Tutor:**

Ing. Dunier Domínguez Mora

**La Habana, junio de 2011.**

**“Año 53 de la Revolución”**

## **Dedicatoria**

*A mis queridos padres, Daniel Díaz y Leticia I. Saez, por ser las personas más importantes en mi vida, por guiarme en todo momento y mostrarme el camino correcto.*

*A ustedes les dedico este trabajo que es el fruto de mi esfuerzo, gracias por brindarme su apoyo y por convertirme en la persona que soy.*

*A mi niño Daniel Lorenzo, por quererme tanto, por llenarme de alegría y por significar mucho para mí, a ti que eres mi mayor tesoro.*

*A mi familia en general, a quienes quiero, por el apoyo que siempre me han brindado, por confiar y estar orgullosos de mí.*

*A mis amigas y más que amigas hermanas, Magaly y Maylin por todos los momentos que hemos vivido juntas.*

*Yelen.*

*A mis padres y mi hermana, por todo el amor, la comprensión y el apoyo que me ofrecen, porque aunque estemos lejos físicamente, nunca dejan de estar a mi lado; porque son lo mejor que tengo en el mundo.*

*A mi mamita preciosa, por apoyarme en todo, por ser la persona que más admiro y quiero en la vida, te dedico este, el primero y más importante de todos mis logros.*

*A mi papá, por estar ahí siempre, por alentarme, guiarme y aconsejarme en cada momento, espero que te sientas orgulloso de mí.*

*A mi hermana Nela, por ser el mejor regalo que me han dado mis padres y la mejor hermana del mundo.*

*A Abraham, por ser el hermano varón que nunca tuve, por su ayuda, apoyo y comprensión, por demostrarme que sólo basta proponérselo para lograr grandes cosas.*

*Pedro.*

## **Agradecimientos**

*Quiero agradecer a mis padres por todo el esfuerzo que han realizado para hacer realidad mis sueños. Por estar junto a mí en los momentos buenos y difíciles, por mimarme y consentirme a diario, por demostrarme todo el amor y el cariño que sienten sin pedir nada a cambio. Simplemente gracias por existir y por ser los mejores padres del mundo. Los Amo.*

*A mi pequeño Daniel Lorenzo, por quererme tanto, por estar siempre con los brazos abiertos esperándome y ser el mejor hermano del mundo. Te Amo.*

*A mis abuelos, tíos, primos por apoyarme y quererme incondicionalmente.*

*A mis amigos especiales Magalys, Maylin y Jorge por todos los momentos que hemos vivido juntos y por ser mi familia estos 5 años.*

*A mi compañero de tesis, Pedro, por ayudarme y sobre todo por ser el mejor compañero de tesis que pude tener.*

*A Miriam Verde, por brindarme tu ayuda en los momentos que la necesité, por ser una excelente persona y por demostrarme que podía contar con tu familia.*

*A mis amigos con los que he compartido un pedazo de mi vida y han estado a mi lado apoyándome para llegar hasta aquí, Yoandy, Yenia, María Luisa, Elizabeth, Rocny, Yudalis, Roberto, Jeans, Ana Lizandra, Alleyne, Jorge Emilio, gracias por los momentos que hemos compartido.*

*A mi tutor Dunier, por ayudarnos mucho durante todo el desarrollo de este trabajo y por guiarnos y aconsejarnos.*

*A todos Muchas Gracias de todo corazón... Yelen.*

## **Agradecimientos**

*A mis padres y mi hermana por darme todo el apoyo y las ganas para seguir adelante y lograr mis metas, porque a ellos les debo en gran medida quien soy y porque para ellos son mis triunfos.*

*A mi abuela, a mi tía Mirtha, a mi hermana Leslie y a toda mi familia en general por la ayuda y el apoyo brindado a lo largo de mis cinco años de carrera.*

*A Abraham, por toda la ayuda que me brindó con la tesis y por demostrarme que no hay malos ni tristes momentos cuando se tiene amigos de verdad.*

*A mi compañera de tesis, Yelen, por ser paciente conmigo, por ayudarme y sobre todo por ser la mejor compañera de tesis que pude tener.*

*A mis hermanos de vida Ariel y Yahíma, por los buenos y malos momentos que pasamos a lo largo de mi carrera estudiantil y porque siempre que los necesité estuvieron a mi lado.*

*A mis amigos de la Universidad, Anibal, Carlos y Jorge, porque cada uno en su momento no escatimó esfuerzos para ayudarme en lo que necesitara.*

*A mi tutor Dunier, por alentarnos desde el inicio y por creer siempre en nosotros.*

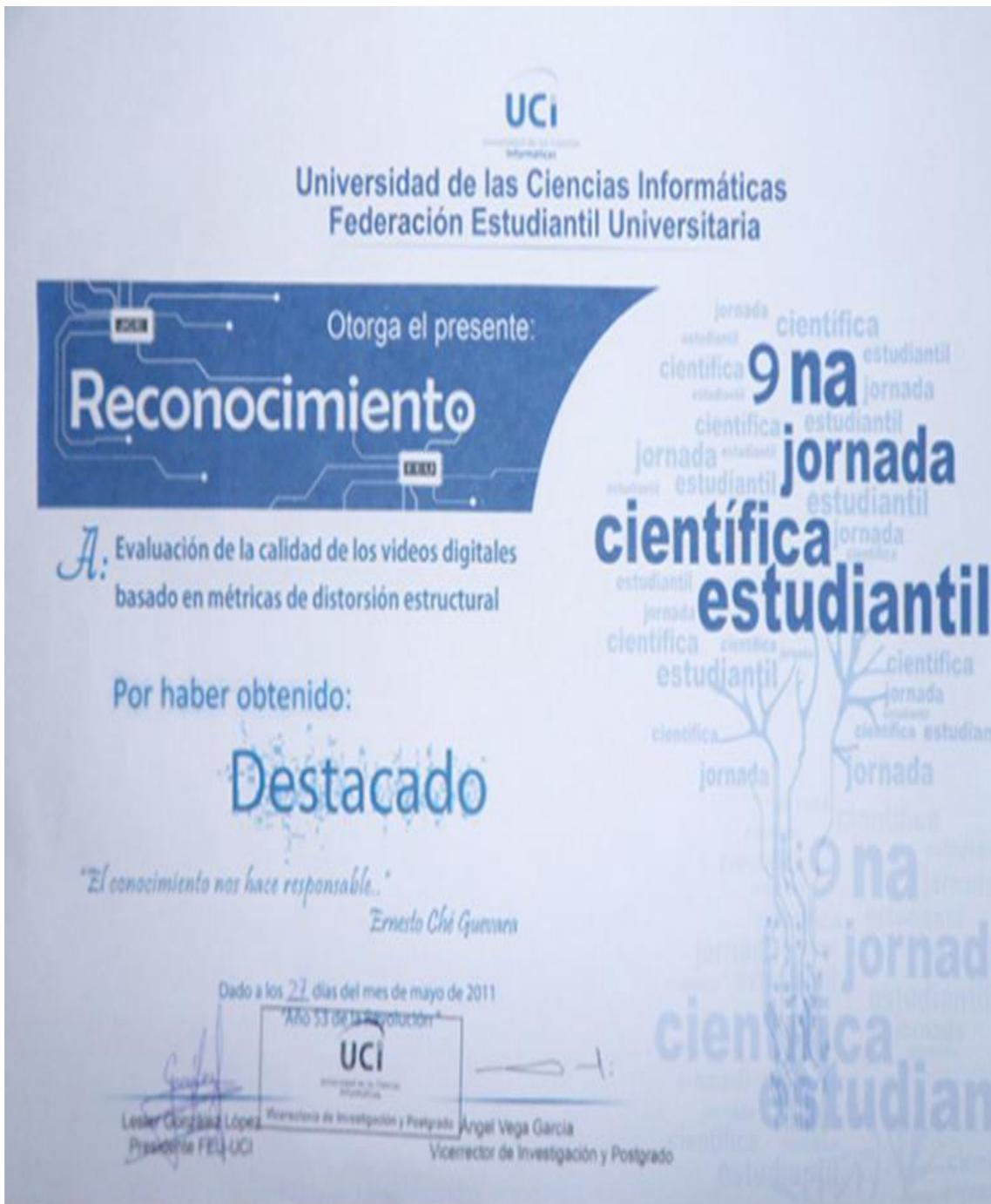
*A mis profesores y amigos Carlitos, Arlen, Dayamí, entre otros muchos que han ayudado a concluir mi trabajo de diploma.*

*A mis compañeros del antiguo 9101 y el actual 6505 por haberme hecho pasar momentos geniales a lo largo de mi carrera.*

*A mis compañeros de proyecto, en especial a los tesistas de PTARTV.*

*Pedro.*

**Reconocimiento otorgado por la Universidad de las Ciencias Informáticas**



## Declaración de autoría

Declaramos que somos los únicos autores del trabajo titulado:

*Evaluación de la calidad de los videos en la plataforma PIARTV basado en métricas de distorsión estructural.*

Autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio con carácter exclusivo. Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Pedro D. Salgado Magaña

[Autor]

---

Yelen Díaz Saez

[Autor]

---

Dunier Domínguez Mora

[Tutor]

### Resumen

La televisión en los últimos años del siglo XX se ha convertido en un importante medio de difusión masiva. En un principio las transmisiones televisivas se realizaban en vivo y a su vez no existía un sistema de grabación eficaz que posibilitara el almacenamiento de los programas. Es así como surgen los sistemas de grabación de video para la televisión. Más adelante con el desarrollo de las computadoras aparece el video digital y junto a esto los avances tecnológicos asociados al procesamiento de imágenes y video digital.

A través del proceso de compresión, almacenamiento y reproducción de video digital, se incorporan con frecuencia una serie de distorsiones las cuales afectan la integridad del video. Es por ello que se convierte en una necesidad de primer orden detectar y cuantificar las degradaciones en la calidad del mismo. En el presente trabajo se muestra la implementación de una herramienta para la evaluación de la calidad de los videos convertidos en la Plataforma de Transmisión Abierta para Radio y Televisión.

Este documento recoge los resultados de la investigación acerca de la teoría de evaluación de calidad de los videos, basada en métricas de distorsión estructural. Además se realiza un análisis profundo de las diferentes tecnologías para su desarrollo. Se describen las funcionalidades de la aplicación a través de los diferentes diagramas realizados.

### Palabras claves

Almacenamiento, Compresión, Distorsiones, Degradaciones, Estructural, Métricas, Plataforma, Procesamiento, Reproducción, Televisión, Video.

### **Abstract**

The TV in the last years of the twentieth century has become an important means of dissemination. Initially TV broadcasts were performed live and in turn there was no effective recording system that would enable the storage of programs. As a result the video recording systems for television arise. Later with the development of computers digital video appears along with technological advances associated with the image processing and digital video.

Through the process of compression, storage and playback of digital video, are often incorporated a series of distortions which affect the integrity of the video. That is why it becomes a prime need to detect and quantify the degradation in quality. The present work shows the implementation of a tool for assessing the quality of the converted videos in the Open Transmission Platform for Radio and Television.

This document contains the results of research on the theory of quality assessment of videos, based on structural distortion metrics. It is also made a thorough analysis of the various technologies for their development. It was described the application functionality exposed through the different diagrams made.

### **Keywords**

Storage, Compression, Distortion, Degradation, Structural, Metric, Platform, Processing, Playback, TV, Video.

## Índice

Introducción .....	14
Capítulo 1. Fundamentación teórica .....	18
1.1 Introducción .....	18
1.2 Conceptos asociados al dominio del problema .....	18
1.2.1 Imagen digital .....	18
1.2.2 Video digital .....	18
1.2.3 Codec .....	18
1.2.4 Codificación de imágenes .....	18
1.2.5 Codificación de video .....	18
1.2.6 Calidad de video digital .....	19
1.2.7 Métricas de calidad en video .....	19
1.3 Objeto de estudio .....	19
1.3.1 Proceso de compresión de imagen y video digital .....	19
1.3.2 Detección de la calidad en imagen y video digital .....	21
1.3.3 Descripción actual del dominio del problema .....	31
1.3.4 Situación problemática .....	31
1.4 Análisis de otras soluciones existentes .....	32
1.4.1 MSU Quality Measurement Tool .....	32
1.4.2 Prototipo de software basado en el Índice de Similitud Estructural de Video (VSSIM) .....	33
1.4.3 Video Quality Measurement (VQM) .....	33
1.5. Conclusiones .....	34
Capítulo 2. Tendencias y tecnologías actuales .....	35
2.1 Introducción .....	35
2.2 Metodologías de desarrollo .....	35
2.2.1 Programación Extrema (Extreme Programming) .....	35
2.2.2 Proceso Unificado de Desarrollo de Software (RUP) .....	36

## Índice

2.2.4 Definición de la metodología a utilizar .....	37
2.3 Lenguaje Unificado de Modelado (Unified Modeling Language UML) .....	38
2.4 Herramientas CASE para la modelación UML .....	38
2.4.1 Rational Rose Enterprise Edition .....	39
2.4.2 Visual Paradigm for UML Enterprise Edition .....	39
2.4.3 Definición de la herramienta CASE a utilizar .....	39
2.5 Lenguajes de programación .....	40
2.5.1 C++ .....	40
2.5.2 Java .....	41
2.5.3 Definición del lenguaje de programación a utilizar .....	42
2.6 Sistema gestor de base de datos (SGBD) .....	42
2.6.1 PostgreSQL 9.0 .....	43
2.7 Qt Creator (IDE) 4.7 como entorno de desarrollo integrado .....	43
2.8 Biblioteca a incluir .....	44
2.8.1 OpenCV 2.1 (Open Source Computer Vision library) .....	44
2.8.2 VTK (Visualization ToolKit) .....	45
2.8.3 Definición de la biblioteca para el procesamiento de imágenes y videos ....	45
2.9 Conclusiones .....	46
Capítulo 3. Presentación de la solución propuesta. ....	47
3.1 Introducción .....	47
3.2 Modelo de dominio .....	47
3.2.1 Conceptos y principales eventos del entorno .....	47
3.2.3 Diagrama del modelo de dominio .....	48
3.2.4 Glosario de términos del dominio .....	48
3.3 Especificación de los requisitos del sistema .....	49
3.3.1 Requisitos funcionales .....	49
3.3.2 Requisitos no funcionales .....	50
3.4 Descripción del sistema propuesto .....	51

## Índice

3.4.1 Descripción de los actores .....	51
3.4.2 Diagrama de casos de uso del sistema.....	52
3.4.3 Descripción de los casos de uso del sistema .....	52
3.5 Conclusiones .....	54
Capítulo 4. Construcción de la solución propuesta .....	55
4.1 Introducción .....	55
4.2 Principios de diseño.....	55
4.2.1 Estándares de la interfaz de aplicación .....	55
4.3 Definición de la arquitectura de software .....	56
4.4 Diagrama de clases del diseño .....	57
4.4.1 Patrones de diseño .....	57
4.5 Diagrama de despliegue .....	62
4.6 Modelo de implementación .....	62
4.6.1 Diagrama de componente.....	62
4.7 Prueba del sistema propuesto .....	63
4.7.1 Pruebas funcionales .....	63
4.7.2 Pruebas unitarias .....	68
4.8 Validación científica de la solución propuesta.....	72
4.9 Conclusiones .....	76
Conclusiones generales.....	77
Recomendaciones .....	78
Referencias Bibliográficas .....	79
Glosario de términos.....	84

## Índice de figuras

Figura 1. Modelo de un sistema general de compresión de video.....	20
Figura 2. Proceso de compresión de videos. ....	21
Figura 3. (a) Imagen original “Goldhill”; (b) Imagen con supresión de contraste; (c) Imagen comprimida JPEG; (d) Imagen desenfocada. ....	26
Figura 4. Flujos, fases e iteraciones de RUP. (21).....	37
Figura 5. Modelo de Dominio. ....	48
Figura 6. Diagrama de casos de uso del sistema.....	52
Figura 7. Arquitectura en capas. (35).....	56
Figura 8. Diagrama donde se evidencia el patrón experto. ....	57
Figura 9. Diagrama donde se evidencia el patrón alta cohesión. ....	58
Figura 10. Diagrama donde se evidencia el patrón controlador. ....	58
Figura 11. Diagrama donde se evidencia el patrón fachada.....	59
Figura 12. Diagrama de clases del diseño del CU “Examinar secuencia a comparar”. 60	
Figura 13. Diagrama de clases del diseño del CU “Chequear calidad”. ....	60
Figura 14. Diagrama de clases del diseño del CU “Emitir reporte de la calidad”. ....	61
Figura 15. Diagrama de clases del diseño. ....	61
Figura 16. Modelo de despliegue del sistema. ....	62
Figura 17. Diagrama de componentes del sistema. ....	63
Figura 18. Fragmento de código correspondiente a la función “Examinar_Secuencia”. .....	69
Figura 19. Grafo de flujo función “Examinar_Secuencia”. ....	69
Figura 20. Fragmento de código correspondiente al método “ValidarVideoEntrada”... 71	
Figura 21. Grafo de flujo función “ValidarVideoEntrada”. ....	71
Figura 22. Representa los videos originales obtenidos de la base de datos del Laboratorio de Ingeniería para Imagen y Video. (40).....	73
Figura 23. Representa las secuencias de videos utilizadas para la validación de la métrica implementada.....	74

## Índice de tablas

Tabla 1. Representación de la densidad de muestreo (RS). (7) .....	29
Tabla 2. Comparación entre XP y RUP.....	38
Tabla 3. Comparación entre Rational Rose y Visual Paradigm. ....	40
Tabla 4. Descripción de los actores del sistema. ....	51
Tabla 5. Caso de prueba #1 para la función “Examinar_Secuencia”.....	70
Tabla 6. Caso de prueba #2 para la función “Examinar_Secuencia”.....	70
Tabla 7. Caso de prueba #1 para la función “ValidarVideoEntrada”.....	71
Tabla 8. Caso de prueba #2 para la función “ValidarVideoEntrada”.....	72
Tabla 9. Caso de prueba #3 para la función “ValidarVideoEntrada”.....	72
Tabla 10. Representa el índice y nivel de calidad utilizando SSIM (precise) para las secuencias analizadas.....	73
Tabla 11. Estimaciones de los algoritmos para las secuencias de evaluación. ....	75

## Introducción

El sector de la informática y las telecomunicaciones se ha convertido en una rama muy importante para la sociedad contemporánea. Esta realidad es posible producto al desarrollo alcanzado por los equipos de cómputo, a partir de la década de los 70 del siglo XX. Desde entonces la radio y la televisión se clasifican entre los medios de comunicación masivos más importantes, su avance se debe entre otros factores, al desarrollo de la informática.

En un principio las transmisiones televisivas se realizaban en vivo y no existía un sistema de grabación que posibilitara el almacenamiento de los programas. Por tales motivos surgen los sistemas de grabación de video analógico para la televisión. Más adelante con el desarrollo de los equipos de cómputo aparece el video digital, que es un tipo de sistema de grabación, que funciona usando una representación digital de la señal de video. Los avances tecnológicos asociados al procesamiento digital de video propiciaron que en poco tiempo esta área alcanzara gran interés económico.

En la era digital que vive la humanidad es común escuchar términos como: internet, correo electrónico, páginas web, servidor, interfaz, compresión, entre otros. Específicamente la compresión de video resulta un tema de vital importancia en la digitalización de los mismos. En esta rama se evidencia la necesidad de la compresión de imágenes y video, donde su uso se hace imprescindible para optimizar los medios de transmisión y almacenamiento.

Una imagen de video no comprimida ocupa aproximadamente 1MB de capacidad. Para lograr un video fluido, se necesita una frecuencia de 25 a 30 imágenes por segundo, lo cual significa un flujo de datos de gran capacidad y poco compatible con el espacio de almacenamiento de los ordenadores personales. Esta dificultad crea la necesidad de emplear técnicas de compresión de datos de video.

La compresión de video e imágenes consiste en un conjunto de algoritmos, métodos y técnicas, ejecutados en sistemas hardware/software para el procesamiento de la información visual. (1) Comúnmente es utilizado en medios donde se necesita producir efectos especiales, mezclar imágenes con gráficas y texto, comprimir la información, estandarizar el formato de video y compatibilizar señales y equipos. A través del proceso de compresión, almacenamiento y reproducción de video digital, se incorporan con frecuencia una serie de distorsiones.

Poder detectar y cuantificar las degradaciones en la calidad de video que se producen en los sistemas, se convierte por tanto en una necesidad de primer orden. En el mundo del procesamiento visual se han establecido métricas para evaluar la calidad en secuencias de videos. Estas métricas se clasifican en dos grandes grupos: subjetivas y objetivas. La evaluación subjetiva se define como la observación que realiza una persona, o un conjunto de estas, para estimar visualmente la calidad de la media. Mientras que las objetivas, mediante fórmulas matemáticas, proporcionan valores numéricos que cuantifican el grado de satisfacción del observador con la imagen procesada. Dentro de esta última clasificación se agrupan las métricas basadas en distorsión estructural, especializadas fundamentalmente en detectar las diferencias estructurales que se producen en la secuencia de video convertida con relación a la original. Este tipo de métrica constituye una solución factible para poder detectar las degradaciones de calidad presentes en un video.

El desarrollo de la informática en Cuba posibilitó que la industria del software ganara una posición importante en la economía. Este salto evolutivo demanda que los productos desarrollados estén dirigidos a las necesidades específicas de los clientes y sean de alta calidad.

Con el surgimiento de la Universidad de Ciencias Informáticas (UCI), se hizo necesaria la creación de la Dirección de Televisión Universitaria (DTU), para apoyar el proceso docente educativo y la formación cultural de la comunidad. Actualmente la DTU publica varios canales a través de la red televisiva por cable y el sitio web [internos.uci.cu](http://internos.uci.cu) (Inter-nos). Este último transmite conferencias, series, películas, documentales, entre otros materiales. La publicación de los materiales audiovisuales tanto por los canales televisivos como por la web, involucra una serie de procesos que tienen un alto contenido de gestión manual. Lo expuesto anteriormente trae asociado errores humanos que conllevan a horas extras de trabajo. Estos motivos dan lugar a la creación de una solución informática para facilitar y mejorar la calidad de transmisión. Dicha solución informática recibe por nombre Plataforma de Transmisión Abierta para Radio y Televisión (PTARTV) perteneciente al Centro Geoinformática y Señales Digitales (GYSED) de la Facultad 6 en la UCI.

La plataforma PTARTV cuenta con doce subsistemas, uno de los cuales es el de Transferencia. Entre las funcionalidades de esta aplicación se encuentran: convertir las medias para su publicación, permitir su visualización para el chequeo de la calidad y el copiado hacia el servidor de medias de la plataforma. En el proceso de

conversión, se presentan errores referentes a la aplicación de un codec<sup>1</sup> específico de video, como pueden ser: el efecto de bloque, el aumento o disminución del contraste/brillo en el video, el mal escalado de la imagen, entre otros. Además, se debe tener en cuenta que algunos de estos errores no son reportados por consola al terminar la codificación, siendo necesario destinar una persona para chequear manualmente la calidad de los videos procesados. Este proceso puede no resolver la problemática de la situación anterior, pues además de ser tedioso e inexacto, trae asociado errores en la revisión de la media. El contexto anterior fundamenta la necesidad de evaluar la calidad del video objetivamente. Existiendo estas dificultades, se requiere la automatización de dichos procesos mediante la implementación de una aplicación para el chequeo de la calidad de las medias (materiales audiovisuales) convertidas en la plataforma PTARTV.

Tomando como base la situación problemática anterior se plantea el siguiente **problema de investigación**: ¿Cómo elevar el nivel de calidad de los videos transmitidos por la PTARTV?

Para dar respuesta al problema planteado se tiene como **objetivo general**: desarrollar un sistema que permita automatizar la evaluación de la calidad de los videos en la plataforma PTARTV, basado en métricas de distorsión estructural.

El **objeto de estudio** del presente trabajo de diploma se centra en la evaluación de la calidad en videos digitales, teniendo como **campo de acción**: la evaluación de la calidad de video en la plataforma PTARTV.

**Se defiende la siguiente idea**: el desarrollo de un sistema para evaluar la calidad de los videos en la plataforma PTARTV, permitirá mejorar la visualización de las transmisiones televisivas.

Con el fin de lograr el objetivo propuesto se plantean las siguientes tareas:

1. Describir la evolución histórico-lógica de los procedimientos y técnicas para la evaluación de la calidad del video digital.
2. Determinar las tendencias y tecnologías actuales más factibles para el desarrollo de la aplicación.
3. Caracterizar procedimientos y técnicas de codificación y decodificación de video digital.

---

<sup>1</sup>Codec es la abreviatura de codificador-decodificador.

4. Caracterizar el procesamiento de imágenes y videos digitales basado en métricas de distorsión estructural.
5. Desarrollar los artefactos y documentación según la metodología de desarrollo seleccionada.
6. Implementar las funcionalidades requeridas.
7. Validar la propuesta de solución.

Para la realización de estas tareas se emplearon diferentes **métodos de investigación**:

### **Métodos Teóricos**

- Analítico - Sintético: permite analizar las teorías y documentos relacionados con el chequeo de la calidad de las medias, posibilitando la extracción de los elementos más importantes que se relacionan con el objeto de estudio.
- Histórico - Lógico: posibilita determinar y comprender la trayectoria histórico-real de los procesos relacionados con la evaluación de la calidad en los videos digitales.
- Modelación: es un método que permite realizar abstracciones simplificadas de la realidad a través de modelos, facilitando comprender el contexto en el que se enmarca la presente investigación.

### **Métodos Empíricos**

- Observación: facilita conocer el panorama real de la situación existente mediante la percepción directa. Con su utilización se determinaron los rasgos imprescindibles en el desarrollo de la solución informática para el chequeo de la calidad de los videos convertidos en la plataforma PTARTV.

## Capítulo 1. Fundamentación teórica

### 1.1 Introducción

En el capítulo se abordan los elementos relacionados con el estudio del estado del arte de los procesos de codificación y evaluación de la calidad en imágenes y videos digitales. Se definen los elementos teóricos que sustentan la investigación y se expone el análisis de las soluciones desarrolladas hasta el momento. Se expresan además, una serie de conceptos que facilitan el entendimiento de la situación problemática.

### 1.2 Conceptos asociados al dominio del problema

#### 1.2.1 Imagen digital

Una imagen digital puede considerarse como una matriz cuyos índices de fila y columna identifican un punto en la imagen y el correspondiente valor del elemento de la matriz identifica el nivel de intensidad de luz en ese punto. (2)

#### 1.2.2 Video digital

El video digital se representa como una secuencia de imágenes (frames), donde además de los parámetros de resolución espacial (número de píxeles) y profundidad (número de bits), hay que considerar un número adecuado de imágenes por segundo que permitan crear la ilusión de movimiento. (3)

#### 1.2.3 Codec

El término codec significa solo una acción: codificar y decodificar archivos multimedia, para poder ser transferidos por la web y reproducidos en los ordenadores. (4)

#### 1.2.4 Codificación de imágenes

La codificación de imágenes se basa en algoritmos para comprimir la cantidad de bit de información que pueda tener una imagen. Utilizando estas técnicas se puede reducir el tamaño de las imágenes hasta cierto grado de compresión sin que se llegue a notar visualmente en la calidad de la representación. (5)

#### 1.2.5 Codificación de video

La codificación de video hace referencia a un proceso en el cual se reduce la cantidad de datos utilizados en la representación del video. Puede ser utilizada para convertir señales de video analógico a señales de video digital. (6)

# Capítulo 1. Fundamentación teórica

## 1.2.6 Calidad de video digital

El término calidad hace alusión a una propiedad o conjunto de propiedades inherentes a un objeto. Extrapolando este concepto a un video, se entendería como la capacidad que tiene una secuencia de video de representar el objeto original, es decir, la exactitud o parecido entre ambos. (7)

## 1.2.7 Métricas de calidad en video

Se define como un conjunto de algoritmos matemáticos, que posibilitan obtener de forma cuantitativa las degradaciones percibidas en la calidad de video. (7)

## 1.3 Objeto de estudio

### 1.3.1 Proceso de compresión de imagen y video digital

La evolución de sistemas multimedia permitió un salto desde la tecnología analógica a la secuencia de imágenes y video digital. Esto se produjo gracias al incremento de la capacidad de procesamiento de la tecnología digital. (1)

El término compresión hace referencia al proceso de reducir la cantidad de datos utilizados para almacenar y transmitir información. Su desarrollo se apoya en que la información, por su naturaleza, no es aleatoria, ya que presenta orden y regularidad. Si el orden y la regularidad se determinan, se pueden eliminar la redundancia e irrelevancia de la información, y representarla con los datos mínimos suficientes para su posterior reconstrucción. (8) Por lo tanto, la necesidad de comprimir un fichero se fundamenta con el hecho de poder almacenar la mayor cantidad de datos en el mismo espacio de memoria de los ordenadores personales. Además este hecho viabiliza la transmisión de datos a través de la red, permitiendo que sea más rápido y eficiente.

Existen dos técnicas de compresión que pueden utilizarse, la compresión con pérdida y la compresión sin pérdida. La primera se aplica cuando la información irrelevante eliminada no es totalmente necesaria y cuando el usuario final no percibe la reducción de la calidad por las mismas razones. Sin embargo, la definición de la segunda no es tan simple, ya que las características dinámicas de los datos generan un gran número de opciones de realización, que hace difícil la medición de los resultados y de la calidad de la compresión. (9)

# Capítulo 1. Fundamentación teórica

## 1.3.1.1 Proceso de compresión en imágenes.

La compresión de la imagen afronta el problema de la reducción de la cantidad de datos necesarios para representar una imagen digital. La base del proceso de reducción de los datos consiste en la eliminación de los datos redundantes. La transformación se aplica antes del almacenamiento o transmisión de la imagen. Posteriormente, la imagen comprimida se descomprime para reconstruir la imagen original o una aproximación de la misma. (9)

La figura 1 representa un sistema de compresión, que consta de dos bloques estructurales distintos: un codificador y un decodificador. Una imagen de entrada  $f(x, y)$  alimenta al codificador que crea un conjunto de símbolos a partir de los datos de entrada. Después de la transmisión a través del canal, la representación codificada alimenta al decodificador donde se genera una imagen  $f'(x, y)$  de salida reconstruida. En general  $f'(x, y)$ , puede ser (o no) una réplica exacta de  $f(x, y)$ . Si lo es, el sistema preserva la información, es decir, está libre de error; si no lo es, el sistema presenta algún nivel de distorsión en la imagen reconstruida. (9)

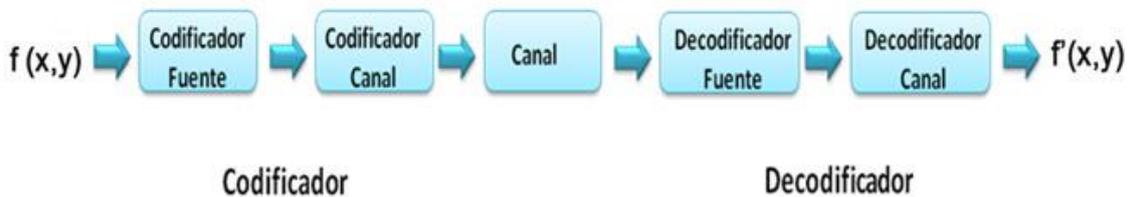


Figura 1. Modelo de un sistema general de compresión de video. (9)

## 1.3.1.2 Proceso de compresión en video

La representación visual mediante secuencia de imágenes continuas introdujo el término video digital. Este es un tipo de sistema de grabación 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. (10)

En la compresión digital de video, se pueden identificar y aprovechar dos tipos básicos de redundancia, la estadística y la psicovisual. La estadística contiene a su vez a la redundancia entre píxeles y la de codificación. Y como una clasificación más reducida y perteneciente al conjunto de la redundancia entre píxeles se encuentra la redundancia espacial y temporal. Esta última, también llamada inter-cuadros, consiste en determinar la correlación estadística entre píxeles de cuadros consecutivos de una

# Capítulo 1. Fundamentación teórica

secuencia de video, mientras que la espacial emplea la misma correlación estadística pero entre píxeles de un cuadro de video específico. La redundancia de codificación busca cancelar la información repetida durante el proceso de compresión de los datos de una secuencia de video. Como última clasificación, la psicovisual se utiliza para eliminar la información visual menos importante para el Sistema Visual Humano (SVH). La compresión de datos se consigue cuando una o varias de estas redundancias se reducen o se eliminan. (1)

La compresión de datos de video hace referencia a un proceso, en el cual se reduce la cantidad de datos utilizados en la representación del video, mientras que la calidad del video reconstruido o recuperado satisface los objetivos de la aplicación de un codec. La figura 2 representa el esquema del funcionamiento de los procesos de compresión de datos del video, durante su transmisión y almacenamiento. (1)

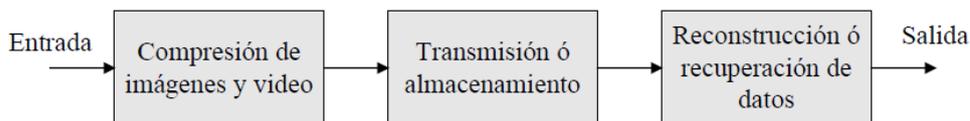


Figura 2. Proceso de compresión de videos. (1)

Durante el proceso de codificación de los sistemas de multimedia, dígase imagen o video, se generan ciertas distorsiones que pueden alterar la fuente procesada hasta niveles perceptibles al SVH. Este último puede detectar algunas anomalías menos graves producto del proceso de conversión y sin embargo puede no detectar otras de un nivel de distorsión mayor. El resultado de este fenómeno visual se justifica en una de las teorías referentes al modelo de visión humana conocida como “modelo multicanal”. Entre las degradaciones más perceptibles para el SVH se pueden encontrar: efecto de bloques (blocking), efecto imagen de base (basis image), desplazamiento de color (color bleeding) y el efecto escalera (staircase effect).

## 1.3.2 Detección de la calidad en imagen y video digital

La detección de la calidad en los sistemas multimedia se hace una necesidad con la existencia de errores en el proceso de conversión, que se traduce en la obtención de medias defectuosas. El desarrollo de la televisión digital y la aparición de diferentes soluciones tecnológicas, exigen nuevas consideraciones de la calidad del video para beneficiar a los telespectadores. Por tanto, poder detectar y cuantificar las degradaciones en la calidad de video que se producen en los sistemas se convierte en una prioridad.

## Capítulo 1. Fundamentación teórica

Para alcanzar un mejor entendimiento de los términos tratados en este epígrafe es preciso comenzar por lo que se entiende como calidad de video e imagen digital. En el mundo del procesamiento visual un problema existente y que aún está sujeto a investigación es la ausencia de una métrica de calidad definitiva, aplicable al proceso de detección de la calidad en video digital.

En los modelos para medir la calidad se puede localizar dos tipos de secuencias, una es considerada la de referencia u original y otra sobre la que se ha efectuado alguna operación de compresión o procesado. En este caso, la medida de calidad de video es una medida de semejanza entre la secuencia original y la distorsionada. (11)

Existen tres tipos de modelos diferentes para poder evaluar la calidad de un video. El primero es aplicable cuando se dispone íntegramente de la referencia original (Referencia Completa). Para el segundo sólo se necesita disponer de cierta información de la señal de video (Referencia Reducida), y el último cuando no se dispone de información alguna (Sin Referencia). Tanto el segundo como el tercer modelo son más complejos de implementar y aún se están realizando estudios teóricos al respecto. (11) Por tal motivo la presente investigación se centra en los sistemas donde se dispone de una secuencia original.

La clasificación de los modelos de medida, se diferencian en dos grupos fundamentales: medias subjetivas (utilizando observadores) y las objetivas (mediante medidas matemáticas).

### 1.3.2.1 Valoración subjetiva de calidad

El término calidad subjetiva de video digital consiste en la revisión de forma manual de las medias convertidas en el proceso de codificación. Esta valoración subjetiva se basa en que un conjunto de personas opinan acerca de su percepción con respecto a una media determinada. La opinión obtenida mediante el Sistema de Medida Subjetivo (MOS por sus siglas en inglés) es la métrica generalmente aceptada como medida de calidad. Por ello, los experimentos subjetivos controlados, se siguen considerando como métodos de medida para la estimación perceptual de la calidad del video.

El MOS constituye la medida subjetiva de mayor difusión para la determinación de la calidad de imagen y video, pero existen otras, entre las que se pueden relacionar: la Escala de Degradación con Doble Estímulo (DSIS por sus siglas en inglés), la Evaluación de Calidad Continua de Estímulo Único (SSCQE por sus siglas en inglés) y la Metodología de Valoración Subjetiva para Calidad de Video (SAMVIQ por sus siglas

## Capítulo 1. Fundamentación teórica

en inglés). Como todo método subjetivo su ejecución conduce a una serie de desventajas obvias: son costosos, difíciles de realizar, e impracticables en aplicaciones de tiempo real. (12)

### 1.3.2.2 Valoración objetiva de calidad

Las métricas de calidad objetivas, mediante fórmulas matemáticas, proporcionan valores numéricos que cuantifican el nivel de satisfacción del observador con la media procesada. Estas métricas pueden proporcionar de forma automática y en un período relativamente corto, una medida de calidad para una imagen o una secuencia de video. El propósito es alcanzar medidas de calidad objetivas con un alto grado de similitud con la calidad subjetiva y así lograr un servicio óptimo en los sistemas de televisión digital. Existen diversas medidas de calidad de video objetiva, desde las que se limitan a calcular las diferencias entre las secuencias originales y comprimidas, hasta las que aplican cálculos complejos basados en el SVH.

Como una primera clasificación se encuentran los modelos de detección de errores, que obtienen una medida relativa a la diferencia de las imágenes entre sí, pudiendo incorporar en mayor o menor medida las características propias del SVH. En un segundo grupo se encuentra una filosofía basada en la distorsión estructural existente entre las secuencias. Finalmente en un tercer grupo se encuentran los modelos híbridos de medida de calidad, en el que se describen algunas métricas que a pesar de no incluir un modelo del SVH tratan de obtener valores que se correspondan con la calidad perceptual. (7)

Las medidas más sencillas para detectar calidad se basan en el cálculo de estadísticas de error numérico entre la imagen distorsionada y la de referencia. Entre las más utilizadas se pueden mencionar el Error Cuadrático Medio (MSE por sus siglas en inglés) y la Relación Señal a Ruido de Pico (PSNR por sus siglas en inglés). El mayor inconveniente que plantean es la relativa falta de correlación con las medidas subjetivas y a su vez con el SVH.

La visión humana es un mecanismo muy complejo que ha sido estudiado durante años. El modo en que el ojo humano percibe el color y el movimiento juega un papel muy importante en la definición de calidad de video. Por ejemplo, debido a la anatomía y a las características físicas del ojo humano, éste es más sensible a los colores rojizos y menos a los tonos amarillos y azules. (13) De esta manera, la pérdida de detalles en las dimensiones del azul y el amarillo, será menos evidente para el ojo

## Capítulo 1. Fundamentación teórica

humano. Estos parámetros tienen menos peso a la hora de diseñar una métrica que permita la evaluación desde la experiencia perceptual.

Existen otros sistemas que buscan una mayor correlación con la distorsión percibida, o tratan de caracterizar el sistema visual humano. A ese modelo se corresponden los métodos de Tektronix/Sarnoff, Modelo de Calidad de Video Digital (DVQ por sus siglas en inglés) y la Métrica de Distorsión Perceptual de la Escuela Politécnica Federal de Lausana (PDM por sus siglas en inglés).

El paradigma de medida de calidad de imagen y video basado en detección de error considera cualquier tipo de distorsión de la escena como cierto tipo de error. Sin embargo, diferentes estructuras de error tendrán efectos distintos en la calidad percibida, por lo tanto, la efectividad de esta aproximación depende de cómo las estructuras de los errores son entendidas y representadas. (7) Constituyendo estas consideraciones posibles errores en dichos algoritmos.

El problema es que el SVH es sumamente complicado y el conocimiento existente sobre el mismo es limitado. Como mismo existen diversas teorías al respecto ninguna de ellas ha sido en su totalidad demostrada, por lo que se siguen realizando trabajos referentes al tema. La mayoría de los enfoques basados en la sensibilidad ante errores son concebidos utilizando el llamado error Minkowski, que es independiente de la estructura de la señal, mediante el uso racional de diferenciación pixel a pixel de la misma. Por lo tanto, la motivación del nuevo enfoque es encontrar una forma de comparar las estructuras de la señal de referencia con relación a la distorsionada.

Un primer aporte para esta teoría lo constituye el libro de Wang, Bovik y Lu “¿Por qué es tan difícil medir la calidad en imágenes? (Why is image quality assessment so difficult?)”, publicado en el 2002.

*“La función principal del sistema visual humano es extraer la información estructural del campo de visión, y el sistema visual humano está altamente adaptado para este propósito. Por lo tanto, una medida de la distorsión estructural podría ser una buena aproximación de la distorsión de imagen percibida”.* (14)

Se considera información estructural aquellos atributos que reflejan la estructura de los objetos en la escena, independiente del nivel de luminancia y contraste de la imagen. Por lo tanto se obtiene una asignación de calidad que separa la medida de distorsión de luminancia, de contraste y estructural. (15)

## Capítulo 1. Fundamentación teórica

Cuando se habla de estructura lo que se quiere decir es que existe una dependencia entre las formas de los objetos de una imagen, que aumenta a medida que dichas estructuras están más cercanas. A diferencia de los métodos basados en el SVH en este método se considera el entorno que rodea al pixel.

La puesta en práctica de esta nueva teoría trae consigo indudables beneficios en el campo del procesado visual. Por ejemplo la mayor diferencia con respecto al proceso de detección de errores, es que la degradación en la imagen es considerada como una pérdida de información estructural percibida en lugar de percepción de errores. Una prueba práctica que puede ejemplificar este planteamiento lo constituye la figura 5, donde la imagen es distorsionada con supresión del contraste global, compresión JPEG y desenfoco. Todas estas imágenes fueron procesadas para obtener un mismo valor de MSE. Curiosamente las imágenes exhiben cualidades visuales completamente diferentes. Esto puede entenderse fácilmente con la nueva filosofía de examinar cómo las estructuras de la imagen se conservan en las imágenes distorsionadas. En la compresión JPEG y el desenfoco, difícilmente se puede observar alguna estructura detallada de la imagen original. Por el contrario, casi todas las estructuras de la imagen con supresión del contraste están muy bien conservadas. (15)



a)

b)

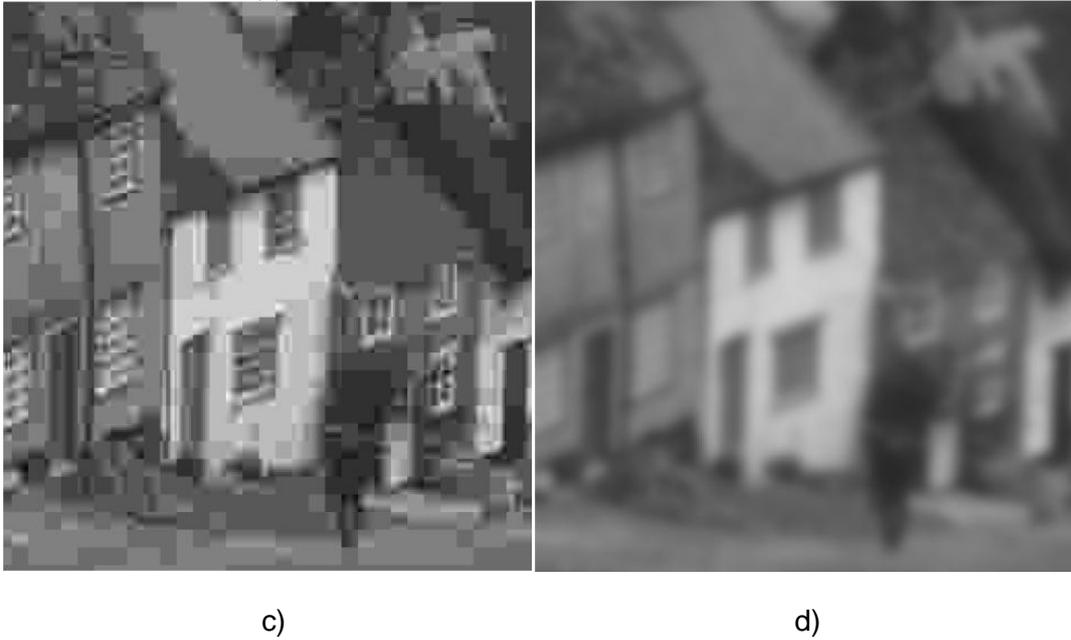


Figura 3. (a) Imagen original "Goldhill"; (b) Imagen con supresión de contraste; (c) Imagen comprimida JPEG; (d) Imagen desenfocada. (15)

Otra diferencia importante es que utiliza una aproximación de arriba hacia abajo (top-down según la bibliografía consultada), simulando la hipótesis del funcionamiento del SVH. Mientras que la filosofía basada en la sensibilidad de error utiliza un enfoque de abajo hacia arriba (bottom-up según la bibliografía consultada), intentando simular la función de cada uno de los componente de relevancia del SVH para luego combinarlos. Es necesario mencionar que la nueva filosofía no tiene intención de resolver los problemas que padece el método basado en detección de errores (por ejemplo, el del "supraumbra" y la "complejidad de las imágenes naturales"). (15)

Los creadores de esta teoría propusieron un algoritmo que calcula el Índice de Similitud Estructural (SSIM por sus siglas en inglés), basado en distorsiones estructurales entre una imagen original (x) y otra de referencia (y). Dicho algoritmo responde a la fórmula:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Donde se define  $\mu_x$  como la media de X,  $\mu_y$  como la media de Y,  $\sigma_x^2$  como la varianza de X,  $\sigma_y^2$  como la varianza de Y y  $\sigma_{xy}$  como la covarianza de X y Y respectivamente. Además  $C_1$  y  $C_2$  son constantes definidas como:

## Capítulo 1. Fundamentación teórica

$C_1 = (K_1L)^2$  y  $C_2 = (K_2L)^2$ . L se define como el rango dinámico de los valores de los píxeles, con  $K_1$  y  $K_2$  constantes pequeñas de forma tal que  $C_1$  y  $C_2$  tengan efecto sólo cuando  $(\mu_x^2 + \mu_y^2)$  o  $(\sigma_x^2 + \sigma_y^2)$  toma valores pequeños. (15)

El índice SSIM satisface las siguientes condiciones:

$$\text{SSIM}(x, y) = \text{SSIM}(y, x);$$

$$\text{SSIM}(x, y) \leq 1;$$

$\text{SSIM}(x, y) = 1$  si y sólo si  $x=y$  (en representación discreta,  $x_i = y_i$  para todo  $i=1,2, \dots, N$ ).

Basados en la nueva filosofía, si se considera que una de las imágenes tiene calidad perfecta, el SSIM da una medida cuantitativa de la distorsión presente en la otra imagen. La distorsión es nula si ambas imágenes son iguales ( $\text{SSIM} = 1$ ) y cuanto menor sea el valor de SSIM mayor es la distorsión. Este algoritmo es aplicable a imágenes estáticas empleando una ventana deslizante de  $8 \times 8$  píxeles. El SSIM se calcula para la región contenida en la ventana, la cual se desplaza desde la parte superior izquierda a la parte inferior derecha de la imagen. El resultado de los cálculos realizados en una imagen queda almacenado en un mapa de índices SSIM, que se considera el mapa de calidad de la imagen en evaluación. El valor global se define como la media del mapa de calidad, es decir, el índice SSIM medio (MSSIM). (15)

Tomando como base esta teoría, se han desarrollado una serie de algoritmos matemáticos que extrapolan el cálculo del índice de similitud estructural, de una imagen a una secuencia de video. A continuación se relacionan algunos de estos:

### Índice de Similitud Estructural en Video (VSSIM por sus siglas en inglés)

VSSIM establece una fórmula matemática para calcular el índice SSIM para cada frame del video, combinando luego dichos valores mediante una suma ponderada para obtener una medida de calidad para todo el video. Para ello se toman un número de ventanas deslizantes ( $R_s$ ) de  $8 \times 8$  píxeles por frame de video de manera aleatoria. De esta forma se reduce significativamente la cantidad de cálculo a realizar. Se aplica el algoritmo de forma independiente a las componentes  $Y$ ,  $C_b$  y  $C_r$  y luego se combinan en un valor local usando una suma ponderada. (15)

Se llama  $\text{SSIM}_{Y_{ij}}$ ,  $\text{SSIM}_{C_b_{ij}}$  y  $\text{SSIM}_{C_r_{ij}}$  al valor de SSIM de la componente  $Y$ ,  $C_b$  y  $C_r$  de la  $j$ -ésima ventana del  $i$ -ésimo frame del video. El índice local está dado entonces como (15):

## Capítulo 1. Fundamentación teórica

$$SSIM_{ij} = W_Y SSIM_{ij}^Y + W_{Cb} SSIM_{ij}^C + W_{Cr} SSIM_{ij}^C$$

Donde los valores considerados fueron  $W_Y = 0,8$ ,  $W_{Cb} = 0,1$  y  $W_{Cr} = 0,1$ . Una vez obtenidos los valores locales, se combinan para obtener un índice de calidad del frame en cuestión, utilizando: (15)

$$Q_i = \frac{\sum_{j=1}^{R_s} w_{ij} SSIM_{ij}}{\sum_{j=1}^{R_s} w_{ij}}$$

Donde  $Q_i$  representa el índice de calidad del  $i$ -ésimo frame, y  $W_{ij}$  es el peso dado a la  $j$ -ésima ventana del  $i$ -ésimo frame. Finalmente, se obtiene el valor de calidad para todo el video (15):

$$Q = \frac{\sum_{i=1}^F W_i Q_i}{\sum_{i=1}^F W_i}$$

Siendo  $F$  el largo en frames del video y  $W_i$  es el peso asignado al  $i$ -ésimo frame. Para elegir los pesos los factores psicológicos son importantes, y aquí entra de nuevo el SVH. De acuerdo con los creadores de este algoritmo se consideraron dos posibles mejoras al método. La primera se basa en que zonas oscuras llaman menos la atención que las claras. Utilizando el valor medio  $\mu_x$  de la componente  $Y$  como estimado de luminancia, se toma el peso local como: (15)

$$w_{ij} = \begin{cases} 0 & \text{si } \mu_x \leq 40 \\ (\mu_x - 40)/10 & 40 < \mu_x \leq 50 \\ 1 & \mu_x > 50 \end{cases}$$

El segundo ajuste tiene en cuenta si existe mucho movimiento en el video. La distorsión percibida depende del movimiento presente en el video. Para esto se calcula el vector de movimiento de cada ventana, buscando la ventana en el frame siguiente que maximice la correlación con la primera. Si  $M_{ij}$  representa el módulo del vector de movimiento de la  $j$ -ésima ventana del  $i$ -ésimo frame, el nivel de movimiento del frame  $i$  se estima como: (15)

$$M_i = \frac{\sum_{j=1}^{R_s} m_{ij}}{R_s K_M}$$

## Capítulo 1. Fundamentación teórica

Donde la constante KM sirve como normalización del nivel de movimiento ( $KM = 16$ ).  
El peso de cada frame es:

$$W_i = \begin{cases} \sum_{j=1}^{R_s} m_{ij} & \text{si } M_i \leq 0,8 \\ (1,2 - M_i)/0,4 & 0,8 < M_i \leq 1,2 \\ 0 & M_i > 1,2 \end{cases}$$

### Índice Modificado de Similitud Estructural en Video (VSSIMm por sus siglas en inglés)

Como su nombre lo indica, el VSSIMm se realiza partiendo del cálculo del índice SSIM y haciendo uso de las técnicas del VSSIM. Para la obtención de este algoritmo es necesario realizar algunos cambios y ajustes a la medida original para la obtención del índice modificado (VSSIMm).

En primer lugar se seleccionan zonas aleatorias de muestreo en bloques de 16x16 píxeles. El tamaño de ventana difiere con el propuesto por los creadores de la medida (8x8 píxeles); se elige este valor para recoger de forma más amplia la estructura de la imagen. Se procesan únicamente una parte de todas las ventanas 16x16 posibles. Tras numerosas pruebas experimentales con el fin de determinar el número de ventanas de muestreo por fotograma, el parámetro  $R_s$  (que representa la densidad de muestreo) se fija por los realizadores de esta métrica por defecto a 0,1. Aunque este parámetro se puede variar en dependencia de la obtención de los resultados, a continuación se muestra una gráfica donde se justifica el valor fijado. (7)

RS	0,1	0,5	1
Tiempo de Ejecución	19 s	143 s	549 s
Variabilidad en el Resultado	8,3 %	2,4 %	0 %

Tabla 1. Representación de la densidad de muestreo ( $R_s$ ). (7)

La fórmula utilizada para el cálculo de esta nueva métrica será la misma que para el VSSIM, teniendo en cuenta las modificaciones vistas anteriormente en el parámetro de  $R_s$ .

# Capítulo 1. Fundamentación teórica

## Borde de Similitud Estructural en Video (VESSIM por sus siglas en inglés)

El SVH es más sensible a detectar distorsiones en el borde de la imagen y según los creadores de la métrica, es la información más relevante a la hora de detectar las distorsiones estructurales. Tomando como base este tipo de información se crea este algoritmo basado en el cálculo del borde de similitud estructural.

Con el fin de estimar las degradaciones en el borde de una imagen, los filtros de detección de bordes se aplican a la estructura original. Inicialmente se obtiene por separado el valor del borde horizontal de la imagen y luego el del borde vertical, utilizando los filtros de Sobel. Estos valores pueden ser estimados mediante la siguiente fórmula:  $g(m, n) = |G_{\text{horizontal}}(m, n)| + |G_{\text{vertical}}(m, n)|$ . (16)

El algoritmo divide el borde de la imagen original y la distorsionada en regiones de 8x8 píxeles, y calcula el valor medio de distorsiones en cada uno. Luego se seleccionan las regiones cuyas distorsiones superan los valores medios de un umbral fijado con anterioridad ( $T=70$ ). (16)

Después de seleccionar las regiones locales, el algoritmo calcula los valores del índice SSIM de cada una y los combina en un valor de calidad para cada una de las imágenes mediante:

$$Q_i = \frac{\sum_{j=1}^{R_i} w_{ij} \times SSIM_{ij}}{\sum_{j=1}^{R_i} w_{ij}}$$

$SSIM_{ij}$  y  $W_{ij}$  denotan el valor de calidad y el factor de ponderación de la  $j$ -ésima región local seleccionada en el marco  $i$ .  $Q_i$  es el valor de calidad del marco  $i$ -ésimo en la secuencia de video. La varianza de cada región, que ha sido calculada en los pasos anteriores se utiliza como factor de ponderación. (16)

La calidad general para la secuencia de video se calcula mediante la fórmula:

$$Q = \frac{\sum_{i=1}^F W_i Q_i}{\sum_{i=1}^F W_i}$$

Donde  $F$  es el número de fotogramas y  $W_{ij}$  el factor de ponderación del  $i$ -ésimo frame. Por lo general se utiliza un solo factor de ponderación, pero sin embargo dicho factor

## Capítulo 1. Fundamentación teórica

se regula de forma tal, que para los frames demasiado suavizados, el número total de determinadas regiones locales sea demasiado pequeño. El factor de ponderación se ajusta como:

$$W_i = \begin{cases} 0 & ES \leq 40 \\ (ES - 40) / 260 & 40 < ES \leq 300 \\ 1 & ES > 300 \end{cases}$$

ES denota el número de regiones locales seleccionadas. (16)

### 1.3.3 Descripción actual del dominio del problema

La UCI surge en el 2002 con el propósito de formar profesionales altamente calificados en el desarrollo de soluciones de software y contribuir a la informatización de la sociedad cubana. Desde sus inicios se prestó especial atención a la formación educacional y cultural de la comunidad universitaria. Por tal motivo se crea la DTU la cual cuenta con 24 canales televisivos y una emisora radial.

El Centro de Geoinformática y Señales Digitales (GEYSED) de la facultad 6, es el encargado de brindar soporte de software a los procesos que se desarrollan en la DTU. Para ello se lleva a cabo el desarrollo del proyecto de investigación y desarrollo Plataforma de Transmisión Abierta para Radio y Televisión (PTARTV). En toda televisora tanto para la transmisión de los canales, como para la publicación en la web de un material de audio o video, se necesita que el fichero original esté codificado en el formato requerido. El subsistema encargado de realizar el proceso de conversión de las medias en la plataforma PTARTV es el Transferencia. Algunas de las funcionalidades que incorpora esta solución informática son: convertir medias de forma automática o manual, previsualizar la conversión, unir y cortar medias y transferirlas hacia el servidor de transmisión.

### 1.3.4 Situación problemática

Aunque el subsistema Transferencia automatiza en gran medida la codificación de los ficheros, el proceso de evaluación de la calidad sigue siendo de forma manual. Para la realización de esta tarea se asigna una persona que es la encargada de visualmente evaluar la calidad de las medias convertidas en la plataforma PTARTV. En este proceso la persona asignada puede no revisar íntegramente la media, provocando que las transmisiones se realicen sin la calidad requerida. Además implica una pérdida de tiempo considerable, si se tiene en cuenta que de quedar defectuosa la media hay que repetir el proceso de conversión.

# Capítulo 1. Fundamentación teórica

## 1.4 Análisis de otras soluciones existentes

Durante la búsqueda y el análisis de aplicaciones dirigidas a la producción de software para evaluar calidad en videos, es posible notar que estos productos son desarrollados principalmente por empresas del primer mundo, que radican, por consiguiente, en los países más desarrollados. La mayoría de estos productos son software privativo y su utilización acarrea gastos muy elevados. Hoy día existen pocas soluciones informáticas en esta rama, donde su principal función es evitar los disímiles errores que se producen en el proceso de conversión de las medias.

Como producto de la consulta y estudio de materiales bibliográficos, han sido identificados los principales software en este sentido. A continuación se relacionan algunas de las características más significativas de estas soluciones informáticas.

### 1.4.1 MSU Quality Measurement Tool

MSU Herramienta de Medición de Calidad de Video (VQMT) es una herramienta de evaluación objetiva de la calidad de video. Examina videos donde utiliza uno como referencia y el otro lo analiza estableciendo comparaciones entre ambos. MSU VQMT permite crear una comparación objetiva de los codec de video y realizar el proceso de análisis de video filtro.

Entre otras de las características de MSU VQMT se puede mencionar que permite la visualización y puesta en práctica de numerosas métricas para evaluar calidad. Al mismo tiempo posibilita calcular los valores de las métricas para cada marco del video y el valor promedio para la secuencia asociada a dicha imagen. Cuenta además con más de 20 formatos de video de apoyo y brinda la posibilidad de obtener archivos CSV (valores separados por comas) como salida. (17)

Los desarrolladores de dicho software concibieron dos versiones del producto, una propietaria y otra de muchas menos restricciones pero con muy pocas funcionalidades. La versión propietaria cuenta con niveles de configuración y precisión aceptables. Pero esta versión por ser privativa no se puede utilizar en el Subsistema de Transferencia de la plataforma PTARTV. El flujo de trabajo en este subsistema no es adaptable a las configuraciones de la versión libre. Debido a que el software cuenta con varias métricas para evaluar calidad, pero a su vez estas métricas utilizan plugins para realizar comparaciones en los videos y la mayoría de estos los trae inhabilitados por completo. Además la arquitectura y composición de este software no permiten agregar funcionalidades para adaptarlo a las necesidades de la plataforma, sin mencionar que

## Capítulo 1. Fundamentación teórica

no se tiene acceso al código fuente. Esta versión libre está bajo la licencia Creative Commons la cual plantea que el software puede ser utilizado para uso personal o de investigación, pero no con fines comerciales. Además fue concebida con la mayoría de los plugins que le permiten evaluar calidad, como inhabilitados teniendo que pagar por su uso. Por estas razones no es posible utilizar esta versión en el subsistema de transferencia de la plataforma PTARTV.

### 1.4.2 Prototipo de software basado en el Índice de Similitud Estructural de Video (VSSIM)

Este prototipo de software permite obtener el índice de calidad general para una secuencia de video. Es un prototipo desarrollado en MATLAB y la elección de este lenguaje se basa en las numerosas funciones predefinidas que incorpora. Además este lenguaje de programación facilita realizar pruebas al instante mediante la programación en línea de comandos, sin necesidad de compilador. Los mensajes de error que este IDE proporciona, son de gran utilidad a la hora de corregir fallos y depurar el código realizado. En el desarrollo de la interfaz gráfica fue de gran utilidad el entorno de programación visual que ofrece MATLAB. (7)

MATLAB es un software que utiliza bibliotecas a las cuales no se tiene acceso. Constituyendo esto una dificultad a la hora de integrarlo a la solución propuesta. Acotando además que el Subsistema Transferencia integrado a la plataforma PTARTV, por sus características y flujo de trabajo no es posible adaptarle este prototipo de software.

### 1.4.3 Video Quality Measurement (VQM)

VQM es otra herramienta que permite evaluar la calidad del video, proporcionando un medio para evaluar objetivamente la calidad del video. Este software realiza el proceso interactivo de fácil utilización de archivos de video, proveyendo dos métodos para realizar mediciones de calidad de video. El primero se basa en obtener del video original y procesado la información necesaria para efectuar el proceso de calidad. Luego el usuario realiza una secuencia de pasos para seleccionar el modelo de calidad y las escalas subjetivas y aplica el cálculo matemático que propone VQM, obteniendo como resultado una relación de calidad entre ambos videos. (18)

En el segundo método se supone que el usuario tenga guardado en la misma máquina donde se encuentra la aplicación los videos originales y procesados, donde más tarde se le aplicara a los mismos un HRC (Circuito ficticio de referencia) como parte del

## Capítulo 1. Fundamentación teórica

proceso de evaluación. Luego de realizar una serie de pasos se selecciona el modelo de calidad de video y las escalas subjetivas, se aplica el cálculo matemático a través de la métrica VQM brindando como resultado un valor de calidad. (18)

### 1.5. Conclusiones

En este capítulo se definieron aspectos fundamentales para entender el entorno en el que se encuentra enmarcada la presente investigación. La descripción histórico-lógica de los procedimientos y técnicas para evaluar calidad en los videos permitió definir la línea base de la investigación hacia resultados satisfactorios. El análisis de las soluciones existentes arrojó como resultado, que las aplicaciones existentes para evaluar la calidad de los videos digitales encontrados no son adaptables ni configurables al flujo de trabajo de la plataforma. Por último, el estudio de las métricas basadas en distorsión estructural permitió arribar a la conclusión que los algoritmos más factibles para su posterior implementación son VSSIMm y VESSIM.

# Capítulo 2. Tendencias y tecnologías actuales

## Capítulo 2. Tendencias y tecnologías actuales

### 2.1 Introducción

A lo largo de este capítulo se lleva a cabo la consideración actual de herramientas y tecnologías que serán empleadas para solucionar el problema de la investigación. Se realiza un estudio de las metodologías de desarrollo de software más representativas, algunos lenguajes de programación, bibliotecas a incluir para el tratamiento de imágenes y videos, así como otros aspectos de importancia. Posterior a este análisis, se definirán cuáles serán utilizados para modelar, implementar y garantizar el correcto funcionamiento del software.

### 2.2 Metodologías de desarrollo

Para llevar a cabo un software es de vital importancia la organización del trabajo. Debe existir un proceso que estructure e integre las múltiples etapas del desarrollo de la aplicación. El equipo de trabajo debe contar con una guía para organizar sus actividades. Es necesario definir los artefactos que deben ser producidos para lograr los objetivos propuestos. Las metodologías de desarrollo de software proveen las guías para poder conocer el camino a recorrer desde las primeras fases de la construcción del software, asegurando la calidad del producto final, así como también el cumplimiento en la entrega del mismo en un tiempo estipulado. (19) En la presente investigación se hace un análisis de las siguientes metodologías: la Programación Extrema (XP) y el Proceso Unificado de Software (RUP).

#### 2.2.1 Programación Extrema (Extreme Programming)

La programación extrema conocida como XP es una metodología ágil de desarrollo de software enfocada en los clientes y los resultados, ya que la regla a seguir es no producir documentos a menos que sean necesarios. La misma está centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software. Promueve el trabajo en equipo, motiva el aprendizaje de los desarrolladores brindando un buen clima de trabajo. XP se basa en una retroalimentación continua entre el cliente y el equipo de desarrollo, existiendo una comunicación fluida entre todos los participantes, claridad en las soluciones implementadas y valor para enfrentar los cambios. Además se define como una metodología adecuada para proyectos con requisitos imprecisos y muy variables, donde exista un alto riesgo técnico.

## Capítulo 2. Tendencias y tecnologías actuales

Entre sus principales características se encuentran: las pruebas y errores, fundamentadas en valores y prácticas. Además presenta pocos artefactos así como roles, centrándose en resolver el problema lo más rápido posible. (20)

XP cuenta con su propio estilo, reduciendo el costo del cambio en todas las etapas del ciclo de vida del sistema. Esta metodología propone la presencia de un representante del cliente durante el tiempo de desarrollo del software, favoreciendo a superar las expectativas.

### 2.2.2 Proceso Unificado de Desarrollo de Software (RUP)

El Proceso Unificado de Desarrollo de Software (RUP por sus siglas en inglés) proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de alta calidad que resuelva las necesidades de los usuarios dentro de presupuestos y tiempos establecidos.

RUP es un marco de procesos altamente configurables para satisfacer necesidades específicas. Implementa las mejores prácticas del desarrollo de software. Es orientada a objetos y utiliza el Lenguaje Unificado de Modelado como lenguaje de representación visual. A su vez define Quién, Cómo, Cuándo y Qué debe de hacerse en el proyecto. Presentando como principales características: (19)

Dirigido por casos de uso: utiliza los casos de uso para el desarrollo de las disciplinas con los artefactos, roles y actividades necesarias. Además estos son la base para la implementación de las fases y disciplinas de RUP.

Iterativo e Incremental: plantea la implementación del proyecto a realizar en iteraciones, por lo que se pueden definir objetivos por cumplir en cada iteración y así completar el proyecto iteración por iteración. Ejecuta una evaluación satisfactoria permitiendo que el proyecto se mueva a la próxima fase y realiza una revisión del ciclo de vida al finalizar la misma.

Centrado en la Arquitectura: define la estructura de las partes más relevantes de un sistema y una arquitectura ejecutable<sup>2</sup> construida como un prototipo evolutivo.

RUP es uno de los procesos más generales de los existentes en la actualidad, debido a que está pensado para adaptarse a cualquier proyecto. La figura 4 representa el

---

<sup>2</sup> Entiéndase como arquitectura ejecutable una implementación parcial del sistema, concebida para demostrar algunas funciones y propiedades de la aplicación.

## Capítulo 2. Tendencias y tecnologías actuales

ciclo de vida del software de RUP que se descompone en cuatro fases secuenciales. En cada extremo de una fase se realiza una evaluación para determinar si los objetivos de la fase se han cumplido. Aunque algunos flujos de trabajo tienen más carga que otros en fases específicas, todos son aplicables dentro de una misma fase. La imagen que a continuación se muestra contempla los flujos y las fases inmersos en esta metodología. (19)

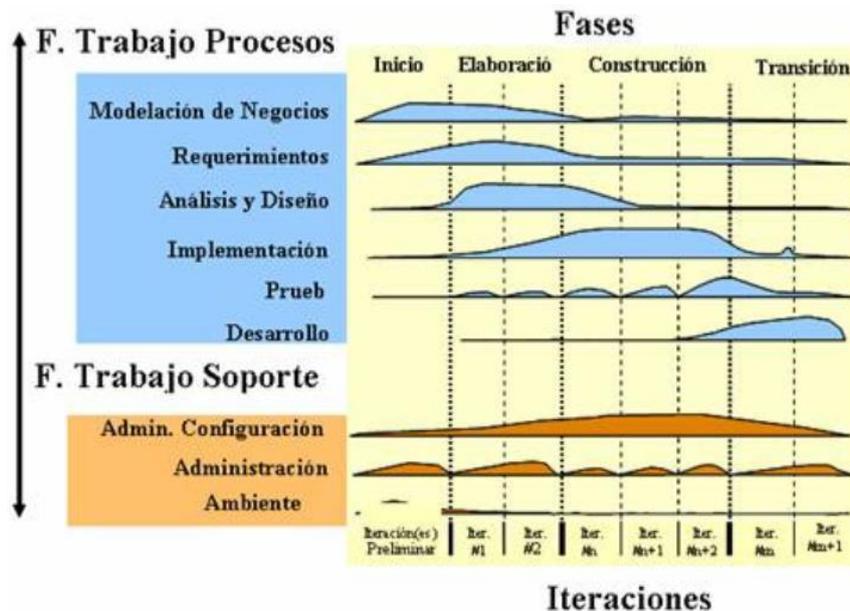


Figura 4. Flujos, fases e iteraciones de RUP. (21)

### 2.2.4 Definición de la metodología a utilizar

Analizando las características de cada metodología de desarrollo de software se puede plantear:

	XP	RUP
Documentación generada	Suficiente	Abundante
Adaptabilidad ante requerimientos variables.	Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Interacción cliente-equipo de desarrollo.	Constante: el cliente es parte del equipo.	Se realiza mediante reuniones.
Equipo de desarrollo.	Pequeño y trabajando en el mismo sitio.	Numeroso y puede estar distribuido.

## Capítulo 2. Tendencias y tecnologías actuales

Capacidad de reutilización.	Baja	Alta
Capacidad de integración.	Baja	Alta

Tabla 2. Comparación entre XP y RUP.

Luego del análisis anterior se puede concluir que la metodología eficaz para el desarrollo de la presente investigación es RUP. Entre las características que justifican la elección realizada se encuentran: genera un volumen abundante de documentación, lo cual es perfecto para el mejor entendimiento del cliente, la capacidad de reutilización es alta, posibilitando que otras aplicaciones se beneficien de las funcionalidades implementadas y por último la capacidad de integración es igualmente elevada, facilitando su alianza con otras soluciones informáticas.

### 2.3 Lenguaje Unificado de Modelado (Unified Modeling Language UML)

UML es un lenguaje visual para especificar, construir y documentar esquemas de los sistemas de software. El mismo contiene diagramas que permiten la modelación de los diferentes componentes que integran el sistema. El Lenguaje Unificado de Modelado es utilizado por metodologías de desarrollo, además presenta una estrecha relación con el Proceso Unificado de Desarrollo de Software (RUP) que es la metodología a utilizar en la presente investigación. (22)

- Otras de las características de UML son:
- Indica lo que supuestamente hará el sistema pero no cómo lo hará.
- Incluye estereotipos como mecanismo de extensibilidad.
- Puede describir cualquier tipo de sistemas en términos de diagramas orientados a objetos.

El desarrollo de sistemas con este lenguaje permite divisar con mayor facilidad las dificultades y dependencias implícitas de la solución informática. Incluye actividades específicas que sirven de guía para saber cómo deben ser las tareas desarrolladas. Además con UML se pueden modelar tanto sistemas de software/hardware como organizaciones del mundo real.

### 2.4 Herramientas CASE para la modelación UML

Las herramientas CASE (del inglés Computer Aided Software Engineering) representan un conjunto de programas y ayudas que dan soporte y asistencia a los analistas, ingenieros de software y desarrolladores, durante todo el ciclo de desarrollo

## Capítulo 2. Tendencias y tecnologías actuales

del sistema. A continuación se analizarán algunas de ellas, seleccionando la más adecuada a utilizar acorde con la metodología de desarrollo de software y el lenguaje de modelado seleccionado.

### 2.4.1 Rational Rose Enterprise Edition

Rational Rose Enterprise Edition es una herramienta CASE que permite crear los diagramas que se van generando durante el proceso de ingeniería en el desarrollo del software. Posibilita especificar, analizar y diseñar el sistema antes de codificarlo. Posee facilidades para la generación de código a partir de diagramas previamente elaborados. Esta herramienta viene acompañada de un sistema de ayuda amigable, al igual que los estereotipos y diagramas que a partir de él se generan, siendo una de las herramientas más técnicas y de más fácil uso.

Es una herramienta propietaria que utiliza como plataforma el sistema operativo Windows, lo que significa que es necesario pagar por su licencia para utilizarla. Conjuntamente no es compatible en el Sistema Operativo GNU/Linux lo cual limita su uso.

### 2.4.2 Visual Paradigm for UML Enterprise Edition

Es una herramienta CASE que soporta múltiples usuarios trabajando sobre el mismo proyecto. Brinda soporte para el modelado y visualización de componentes necesarios durante todo el ciclo de vida del software. Constituye una plataforma para el desarrollo de sistemas con bajos costos y buena calidad. Permite representar todo tipo de diagramas UML para las distintas fases como la captura de requisitos, análisis, diseño e implementación. Esta herramienta tiene la capacidad de generar código y realizar ingeniería inversa con diferentes lenguajes de programación orientados a objetos. Posibilita la integración con diferentes entornos de desarrollo integrados y otras herramientas CASE, brindando facilidades para la exportación e importación de componentes. Esta herramienta de modelado ha ganado gran potencial ya que es utilizada para el desarrollo de sistemas en plataformas libres y a pesar de su licencia comercial su versión *Community* posee licencia gratuita. Es multiplataforma, diseñada para el trabajo tanto en Windows como en Linux.

### 2.4.3 Definición de la herramienta CASE a utilizar

Analizando las características de las herramientas CASE presentadas ha sido posible notar que ambas son eficientes en la modelación de una solución informática. A

## Capítulo 2. Tendencias y tecnologías actuales

continuación se establece una breve comparación entre ambas para seleccionar la más idónea a implementar.

	Rational Rose Enterprise Edition.	Visual Paradigm for UML Enterprise Edition.
UML	Modela todos sus diagramas.	Modela todos sus diagramas.
Ingeniería inversa.	Sí	Sí
Sistemas Operativos que lo soportan.	Windows	GNU/Linux y Windows
Licencia	Es una herramienta propietaria.	Tiene una versión gratuita.

Tabla 3. Comparación entre Rational Rose y Visual Paradigm.

Como conclusión del análisis anterior se propone utilizar Visual Paradigm for UML Enterprise Edition v6.4 ya que soporta que trabajen en un mismo proyecto múltiples usuarios. Esta herramienta está orientada al Lenguaje Unificado de Modelado (UML), influyendo en esto la posibilidad que ofrece su versión *Community* con su licencia gratuita. Presenta facilidad para el trabajo colaborativo y sus características multiplataforma le permiten ejecutarse tanto en Windows como para las diferentes distribuciones de Linux.

### 2.5 Lenguajes de programación

Los lenguajes de programación han sido diseñados para poder escribir instrucciones parecidas a un lenguaje humano, que las computadoras pueden convertir en código binario a través de los denominados compiladores. Siendo los lenguajes un conjunto de reglas, símbolos, notaciones y caracteres que permite al programador poder expresar el procesamiento de datos y sus estructuras en la computadora.

#### 2.5.1 C++

C++ en su inicio era una extensión del lenguaje C y se comenzó a desarrollar en 1980. En la actualidad es un lenguaje versátil, potente y general. El C++ mantiene las ventajas del C en cuanto a riqueza de operadores y expresiones, flexibilidad y eficiencia. Este lenguaje es orientado a objetos y orientado a algoritmo. El código es

## Capítulo 2. Tendencias y tecnologías actuales

portable y puede ejecutarse en cualquier máquina y bajo cualquier sistema operativo. (23)

Principales ventajas:

- Versatilidad: C++ es un lenguaje de propósito general, por lo que se puede emplear para resolver cualquier tipo de problema. (23)
- Eficiencia: C++ es uno de los lenguajes más rápido en cuanto a tiempo de ejecución. (23)
- Herramientas: cuenta con una gran cantidad de compiladores, depuradores, librerías entre otros. (23)
- Portabilidad: es un lenguaje estandarizado y un mismo código fuente se puede compilar en varias plataformas. (23)

Desventajas:

- C++ es complicado y se produce un ejecutable válido sólo para una plataforma concreta. Es híbrido, inseguro y no es apropiado para la web. (23)

### 2.5.2 Java

Java fue diseñado e implementado por la empresa Sun Microsystems, en un principio fue pensado para programar algoritmos sencillos. A partir de los años 1990 comienza el diseño de este nuevo lenguaje de programación, obteniendo como resultado un lenguaje poderoso y confiable, capaz de ejecutarse en todo tipo de procesadores. (24)

Java es orientado a objetos, el cual puede ser ejecutado en múltiples plataformas. Cuenta con una arquitectura neutra y con una buena seguridad, además de ser interpretado, robusto y de alto desempeño.

Otras características de Java son:

Combina aplicaciones o servicios que usan este lenguaje para crear servicios o aplicaciones totalmente personalizados. (24)

Permite el desarrollo de potentes y eficientes aplicaciones para teléfonos móviles, procesadores remotos, productos de consumo de bajo costo y prácticamente cualquier tipo de dispositivo digital. (24)

Java es un lenguaje de hilos múltiples, proporcionando soporte para varios hilos de ejecución, llamado también proceso de peso ligero que pueden manejar diferentes

## Capítulo 2. Tendencias y tecnologías actuales

tareas. Un beneficio de gran importancia de los hilos múltiples es que se mejora el desempeño interactivo de las aplicaciones gráficas para el usuario. (24)

Java es un lenguaje que posee un intérprete llamado maquina virtual, principalmente diseñado para una arquitectura de hardware. Esto a la hora de procesar imágenes y videos por computadora es un problema porque al tener que interpretar el código hace el proceso mucho más lento que C. La máquina virtual de Java es lenta y siendo así, aún no puede competir con el código nativo de C++ en cuanto a tiempos de ejecución se refiere. (25)

### 2.5.3 Definición del lenguaje de programación a utilizar

Luego del análisis de los lenguajes de programación mencionados previamente, se selecciona C++ a utilizar en la implementación de la aplicación. Este lenguaje posee grandes potencialidades además de ser compatible en varias plataformas incluyendo Linux. Incorpora entre sus características una mayor velocidad de ejecución en comparación con otros lenguajes como Java. A continuación se exponen las principales ventajas:

Datos de abstracción: los detalles de implementación están ocultos a otros desarrolladores, lo que permite mantener y mejorar los programas de carácter más fácil. (23)

Programación genérica: hace referencia a una especie de abstracción en el proceso de desarrollo de software, se centra en la construcción de algoritmos que es independiente de un determinado tipo o estructura de datos. (23)

### 2.6 Sistema gestor de base de datos (SGBD)

Un SGBD es una herramienta que permite crear, administrar, almacenar y organizar la información contenida en una base de datos, asegurando su confidencialidad, integridad y control de su seguridad. Está compuesto de un conjunto de programas no visibles al usuario final. Proporciona una interfaz entre los datos, los programas que los manejan y los usuarios finales. (26)

## Capítulo 2. Tendencias y tecnologías actuales

### 2.6.1 PostgreSQL 9.0

El SGBD relacional orientado a objetos conocido como PostgreSQL está derivado del paquete Postgres escrito en Berkeley, distribuido bajo licencia BSD<sup>3</sup>. Con más de una década de desarrollo PostgreSQL ha demostrado ser un gestor de bases de datos de código abierto muy avanzado, ofreciendo control de concurrencia multiversión entre sus características (MVCC). Soportando casi todo tipo de sintaxis SQL (incluyendo subconsultas, transacciones, tipos y funciones definidas por el usuario). Cuenta también con un amplio conjunto de enlaces con lenguajes de programación como pueden ser C, C++, Java, Perl, TCL y Python.

La utilización de este SGBD en la PTARTV obedece al propósito de la elaboración de un sistema con gran robustez y alto nivel de escalabilidad. Trabajar con este gestor es factible ya que es muy rápido, tiene buenas utilidades de administración, es confiable y fácil de usar tanto para volúmenes de datos grandes como pequeños. Consecuentemente tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. (27)

### 2.7 Qt Creator (IDE) 4.7 como entorno de desarrollo integrado.

Qt es una biblioteca multiplataforma para desarrollar interfaces gráficas de usuario y también para el desarrollo de programas sin interfaz gráfica como herramientas de la consola y servidores. Es producido por la empresa de Nokia distribuido bajo los términos de GNU Lesser General Public License.

Qt es utilizada en KDE, un entorno de escritorio para sistemas como GNU/Linux o FreeBSD, entre otros. Qt utiliza el lenguaje de programación C++ de forma nativa, adicionalmente se le pueden incorporar otros lenguajes de programación. Funciona en las principales plataformas, y tiene un amplio apoyo. El API de la biblioteca cuenta con métodos para acceder a bases de datos mediante SQL, así como uso de XML, gestión de hilos, soporte de red y una API multiplataforma unificada para la manipulación de archivos. Cuenta además con varias estructuras de datos tradicionales. QT es software libre y de código abierto. (28)

Principales características de Qt Creator:

---

<sup>3</sup> Licencia de software otorgada principalmente para los sistemas (*Berkeley Software Distribution*) BSD. Es una licencia de software libre, tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público, permite el uso del código fuente en software no libre.

## Capítulo 2. Tendencias y tecnologías actuales

- Ayuda sensible al contexto integrado
- Posee también una GUI integrada y diseñador de formularios
- Soporta los lenguajes: C#/.NET, Python, PyQt y PySide, Ada, Pascal, Perl, PHP y Ruby
- Depurador visual
- Resaltado y autocompletado de código
- Soporte para refactorización de código
- Se visualizan los datos en 3D con una estrecha integración con OpenGL
- Permite avanzadas interfaces de usuario de un amplio conjunto de componentes de interfaz gráfica de usuario estándar y personalizables
- Entrega de alto rendimiento de las aplicaciones a través de plataformas múltiples

### 2.8 Biblioteca a incluir

El uso de bibliotecas proporciona un marco de trabajo de alto nivel para el desarrollo de aplicaciones. Posibilita el procesamiento y visualización de imágenes, estructura de datos y el análisis de dichas estructuras. Cuando se utiliza un IDE como entorno de desarrollo, en este caso QT Creator, el uso de bibliotecas adicionales incorpora funciones que viabilizan el procesamiento de imágenes para la implementación del sistema.

#### 2.8.1 OpenCV 2.1 (Open Source Computer Vision library)

OpenCV es una biblioteca de código abierto que proporciona un gran número de funciones para el procesamiento de imágenes. Permite a los programadores crear aplicaciones poderosas en el dominio de la visión digital. Ofrece muchos tipos de datos de alto nivel como árboles, gráficos, matrices, entre otros. Incluye funcionalidades como: realizar operaciones básicas en matrices, procesado de imágenes, análisis estructural, análisis de movimiento, reconstrucción 3D, calibración de la cámara, segmentación y reconocimiento de objetos, entre otras.

Esta biblioteca implementa gran variedad de herramientas para la interpretación de la imagen. Además de incluir operaciones primitivas como estadísticas de la imagen y filtrado, las cuales son utilizadas en la implementación. Los algoritmos están basados en estructuras de datos muy flexibles. Brinda funciones que posibilitan convertir una imagen en diferentes formatos de color. Genera imágenes de movimiento MHI (Motion History Images) para determinar dónde hubo movimiento y en qué dirección. (29)

## Capítulo 2. Tendencias y tecnologías actuales

### 2.8.2 VTK (Visualization Toolkit)

VTK es una biblioteca de dominio público, distribuido bajo el modelo Open Source, para computación gráfica en 3D, procesamiento de imágenes y visualización. El diseño de esta biblioteca está fuertemente basado en objetos.

El modelo grafico en VTK está a un alto nivel de abstracción. Lo que significa que facilita la creación de gráficos y aplicaciones de visualización. Las aplicaciones desarrolladas con esta librería pueden escribirse directamente en C++, Java o Python. Incluye técnicas de modelado avanzadas como el modelado implícito, reducción de polígonos, el aplanado de mallas, recorte y contorneando. Además integra decenas de algoritmos de imágenes para mezclar procesamiento en 2D y 3D. (30)

Algunas de sus características gráficas son:

Capacidades de Graficación y Render: integra numerosas funcionalidades para desplegar primitivas geométricas. Incluye algoritmos para la visualización de campos escalares, campos vectoriales, visualización de tensores, mapeo de texturas, así como métodos volumétricos para la representación de superficies y volúmenes. Cuenta también con una amplia gama de rutinas para manipulación de imágenes, lo que permite incluir imágenes 2D en las escenas 3D en formas de mapas, texturas o imágenes de fondo.

Requerimientos de hardware y software: la instalación de esta librería requiere un soporte para OpenGL. Esta desarrollada completamente en C++ y posee las interfaces para utilizar las rutinas desde Tcl, Python y Java.

### 2.8.3 Definición de la biblioteca para el procesamiento de imágenes y videos

Definir la biblioteca idónea para contribuir al desarrollo de la aplicación es un tema de vital importancia, si se tiene en cuenta que su utilización es para facilitar la implementación y no para añadirle funcionalidades innecesarias. La utilización de VTK requiere el uso de herramientas demasiado potentes para el trabajo con videos, que le agregarían un elevado costo computacional a la aplicación. Además esta biblioteca está más especializada en trabajar con modelos 3D que con imágenes y su instalación conllevaría a la inclusión de herramientas innecesarias para el proyecto. Sin embargo OpenCV incorpora estructuras de la imagen que facilitan su manipulación digital. Incluye funciones que posibilitan convertir la imagen a diferentes formatos de color y por último cuenta con filtros para simular distorsión en la imagen. Las características

## Capítulo 2. Tendencias y tecnologías actuales

mencionadas referentes a la utilización de la biblioteca OpenCV contribuyeron a determinarla como principal candidata para el desarrollo de la aplicación.

### 2.9 Conclusiones

Las principales metodologías para el desarrollo de software posibilitaron determinar a RUP como metodología a utilizar en la presente investigación. La cual está basada en la orientación a objetos y la modelación visual usando UML, lo que permite incorporar al proceso de desarrollo de software un mejor control de los requerimientos y cambios. Además de ser reutilizable y poseer un alto nivel de integración. Se eligió C++ y Qt por sus eficientes tiempos de ejecución. Se estableció OpenCV por sus facilidades para el tratamiento de imágenes para lograr de manera óptima el objetivo general del presente trabajo de diploma.

## Capítulo 3. Presentación de la solución propuesta

### Capítulo 3. Presentación de la solución propuesta.

#### 3.1 Introducción

En el presente capítulo se abordarán temas relacionados con la propuesta de solución presentada, apoyados en los artefactos generados por la metodología de desarrollo RUP. Se presenta el Modelo de Dominio del problema para facilitar la comprensión de los conceptos básicos asociados, así como los requerimientos identificados. Además se procederá a la modelación del sistema y sus especificaciones.

#### 3.2 Modelo de dominio

Un modelo de dominio es una representación visual de los conceptos u objetos del mundo real más significativos para un problema o área de interés. En el UML se ilustra con un grupo de diagramas de estructura estática donde no se define ninguna operación. La elección de modelo conceptual ofrece la ventaja de marcar una unión en los conceptos del dominio, no en las entidades del software.

Este modelo es un artefacto, construido con las reglas de UML durante la fase de concepción del sistema. Está compuesto por uno o más diagramas de clases y no contiene conceptos propios de un sistema de software sino de la propia realidad física. (31) Puede utilizarse para capturar y expresar el entendimiento obtenido en una determinada área. Los analistas lo utilizan como un medio para comprender el negocio a informatizar por el sistema y es empleado fundamentalmente cuando los flujos de información son indefinidos.

##### 3.2.1 Conceptos y principales eventos del entorno

Una **televisora** es la entidad encargada de la transmisión y difusión de la información audiovisual, donde se aplican procesos de conversión a las medias publicadas, procedentes de un servidor de medias o de una transmisión en vivo.

La plataforma cuenta con un servidor de **medias**, que contiene películas, documentales, videos, entre otros materiales audiovisuales, a los que se les aplica procesos de conversión para su posterior publicación. PTARTV como solución informática implementa 12 subsistemas, entre los que se encuentra el de transferencia. Este último como se ha expuesto con anterioridad engloba numerosos procesos, como por ejemplo los **procesos de conversión**, durante el cual se obtienen las **medias procesadas**.

## Capítulo 3. Presentación de la solución propuesta

Durante este proceso pueden aparecer varios **errores de conversión** que son el resultado de aplicar un codec específico de video. Estos errores son detectados por los **procesos de calidad** que son los encargados de evaluar la calidad mediante una secuencia de pasos, brindando un reporte de la misma. Para el buen desempeño de esta evaluación se emplean **métricas basadas en distorsión estructural**. Entiéndase por esto como una serie de algoritmos matemáticos que calculan el índice de similitud estructural existente entre dos imágenes. Como parte de esta métrica pero extendida a una secuencia de video se encuentra el algoritmo **VSSIMm**.

### 3.2.3 Diagrama del modelo de dominio

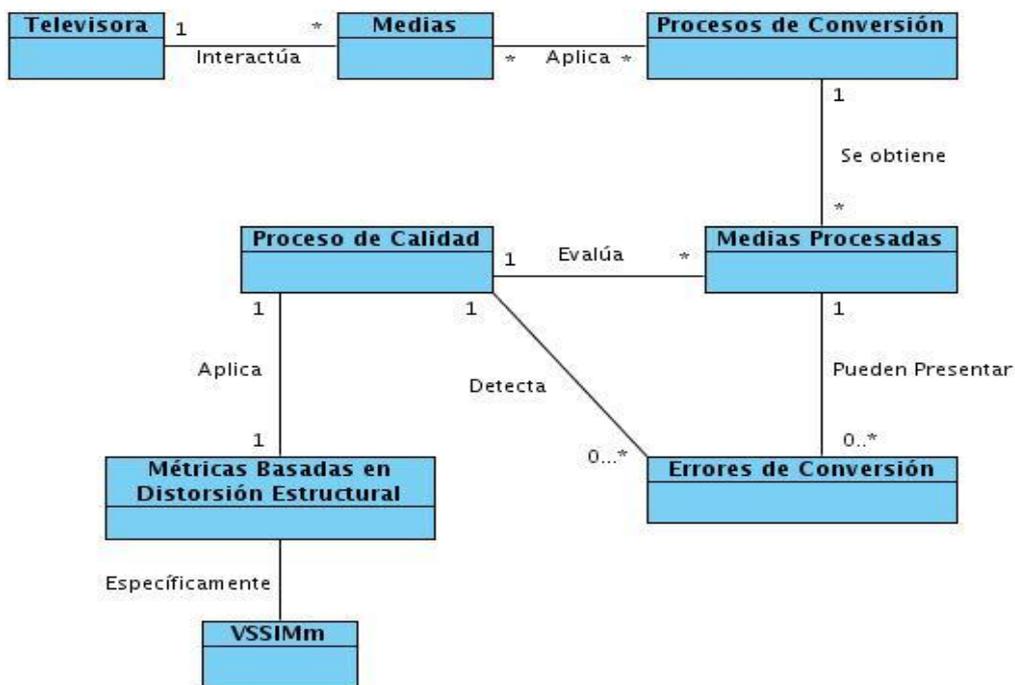


Figura 5. Modelo de Dominio.

### 3.2.4 Glosario de términos del dominio

El siguiente glosario de términos ayuda a una mejor comprensión del diagrama del dominio.

**Televisora:** es la entidad encargada de la transmisión de canales y procesamiento de videos digitales para la difusión y transmisión de la información audiovisual.

**Medias:** son todos los materiales audiovisuales tratados en la plataforma PTARTV.

**Proceso de Conversión:** es el proceso mediante el cual se convierten a un determinado formato de video las medias de la plataforma PTARTV.

## Capítulo 3. Presentación de la solución propuesta

**Medias Procesadas:** son todos los materiales audiovisuales obtenidos como resultado de los proceso de conversión en la plataforma PTARTV.

**Errores de Conversión:** son los errores que se producen durante el proceso de conversión y que provocan que las medias procesadas se encuentren defectuosas y con ello que su calidad no sea óptima.

**Procesos de Calidad:** son los encargados de evaluar la calidad mediante una secuencia de pasos, brindando un reporte de la misma.

**Métricas Basadas en Distorsión Estructural:** son una serie de algoritmos matemáticos que cuantifican las diferencias estructurales presentes en una imagen o secuencia de video.

**VSSIMm:** algoritmo que calcula el Índice de Similitud Estructural (SSIM por sus siglas en inglés) entre dos secuencias de video.

### 3.3 Especificación de los requisitos del sistema

El término requisito se define como una condición o capacidad que debe estar presente en un sistema o componente de un sistema, para satisfacer un contrato estándar entre el cliente y el equipo de desarrollo, especificación u otro documento formal. (32)

Los requerimientos son los que permiten guiar el desarrollo de un software hacia un sistema que cumpla con las exigencias de los clientes. Para el cumplimiento de esta tarea, se ha tomado como punto inicial las necesidades de los usuarios finales. A continuación se mencionan a través de requerimientos funcionales y no funcionales, propiedades o cualidades que el sistema debe ser capaz de realizar.

#### 3.3.1 Requisitos funcionales

**RF1** Seleccionar el tipo de evaluación

El sistema cuenta con dos tipos de evaluación “Evaluación Manual” y “Evaluación Automática”. Donde el sistema brinda la posibilidad al usuario de seleccionar el tipo de evaluación de calidad que desea realizar.

**RF2** Examinar secuencias a comparar

El sistema brinda la posibilidad al usuario de seleccionar la secuencia de video original y la secuencia procesada para establecer la comparación.

## Capítulo 3. Presentación de la solución propuesta

### RF3 Aplicar filtro de muestreo

El sistema debe ser capaz de verificar que el usuario seleccione el número de ventanas de muestreo para realizar la evaluación de la calidad.

### RF4 Evaluar calidad

El sistema debe ser capaz de evaluar la calidad de la secuencia procesada con respecto a la secuencia original verificando cada frame del video.

### RF5 Emitir reporte de la calidad

El sistema debe generar el resultado de la evaluación de la calidad realizada emitiendo un reporte de la misma.

### 3.3.2 Requisitos no funcionales

Los requisitos no funcionales se clasifican en diversas categorías, las cuales hacen que el producto sea confiable, seguro, rápido y usable. Dichos requisitos son propiedades o cualidades que el producto debe de tener. (21)

**Requisitos de usabilidad:** el sistema debe brindar facilidades para que pueda ser usado por personas con conocimientos básicos en el manejo de computadoras.

**Requisitos de ayudas y documentación en línea:** el sistema debe presentar un manual de usuario donde indique al usuario los pasos que tiene que seguir para determinar la calidad de una media.

**Restricciones en el diseño y la implementación:** se debe incluir la librería OpenCV para el tratamiento con imágenes.

**Requisitos de portabilidad:** el sistema se realizará para un funcionamiento en sistemas operativos como GNU/Linux y/o Windows, garantizando de esta manera un producto multiplataforma.

**Requisitos de apariencia o interfaz externa:** la aplicación permitirá realizar un fácil manejo, posibilitando un buen desempeño en la ejecución de sus funciones. También debe contar con una interfaz sencilla y amigable.

**Requisitos de disponibilidad:** el sistema debe permanecer disponible las 24 horas del día y 7 días a la semana.

## Capítulo 3. Presentación de la solución propuesta

**Requisitos de confiabilidad:** la información manejada o generada por el sistema debe estar protegida del acceso no autorizado y de divulgación a terceras personas.

**Requisitos de licencia:** el desarrollo del sistema tanto como el de la plataforma se realiza con herramientas libres.

**Requisitos legales, derecho de autor y otros:** los derechos de autor y otros están establecidos por la empresa comercializadora de productos, ALBET S.A y la entidad desarrolladora de software UCI.

**Requisitos de hardware:** el ordenador donde establezca la aplicación debe cumplir con las siguientes características:

- Requisitos Mínimos: Procesador: Core 2 Duo a 2GHz, Memoria: 1 GB, Disco Duro: 2 GB libre, Tarjeta de red Gigabyte Ethernet 1 Gbps.

### 3.4 Descripción del sistema propuesto

Luego de realizar el levantamiento de requisitos del sistema se describen los casos de uso para la siguiente etapa de construcción. Los requisitos de software se agrupan según sus funcionalidades, conformando así los casos de uso. Estos constituyen un conjunto de instancias, donde cada instancia es una secuencia de acciones que lleva a cabo un sistema para obtener un resultado palpable para un actor concreto. (21) El comportamiento de un sistema específico se describe mediante los casos de uso, donde se representa de manera secuencial las acciones más significativas que realiza un actor en intercambio con el sistema.

#### 3.4.1 Descripción de los actores

Se define como un actor los roles que puede tener un usuario, máquinas, hardware u otros sistemas, que interactúan con un sistema para así intercambiar datos. Un actor no es parte del sistema en desarrollo, es un agente externo que intercambia con el mismo para lograr un resultado esperado. (21)

Actores	Descripción de los actores
Conversor	Persona encargada de procesar las medias haciendo uso del subsistema transferencia.

Tabla 4. Descripción de los actores del sistema.

## Capítulo 3. Presentación de la solución propuesta

### 3.4.2 Diagrama de casos de uso del sistema



Figura 6. Diagrama de casos de uso del sistema.

### 3.4.3 Descripción de los casos de uso del sistema

Los casos de uso se utilizan para capturar el comportamiento deseado del sistema en desarrollo, sin la necesidad de especificar cómo se implementa dicho comportamiento. Brindan un medio para que los usuarios del sistema, expertos del dominio y los desarrolladores lleguen a una visión común del sistema. Además estos son el componente clave para el modelado, lo que permite ilustrar las funcionalidades de un sistema y la relación con el actor para llegar al cumplimiento de un objetivo. Los casos de uso describen procesos de inicio a fin, siguiendo una secuencia de eventos que se requieren para realizarlos. A continuación se describen los casos de usos que conforman el sistema. (21)

#### 3.4.3.2 Chequear calidad

<b>Caso de Uso</b>	Chequear calidad
<b>Actores</b>	Conversor
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de que el usuario pueda seleccionar el número de ventanas por el que desea muestrear y calcular el índice de calidad del video procesado.
<b>Resumen</b>	El caso de uso se inicia cuando el usuario accede a la interfaz principal para seleccionar el número de ventanas por el que desea muestrear. Termina con el cálculo del índice de calidad del video procesado.
<b>Precondiciones</b>	Debe haberse examinado las secuencias a comparar.

### Capítulo 3. Presentación de la solución propuesta

<b>Referencias</b>	RF 3, 4
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El caso de uso se inicia cuando el usuario accede a la interfaz para seleccionar el número de ventanas por el que desea realizar el muestreo.	1.1 El sistema habilita la posibilidad de seleccionar el número de ventanas por el que desea muestrear y los botones “Chequear” y “Cancelar”.
2. El usuario selecciona el número de ventanas por el que desea realizar el muestreo y presiona el botón “Chequear”.	<p>2.1 El sistema verifica que se haya introducido el número de ventanas.</p> <p>2.2 El sistema descompone las secuencias original y procesada en imágenes.</p> <p>2.3 El sistema divide cada imagen en el número de bloques de 16 x 16 pixeles especificadas por el usuario.</p> <p>2.4 El sistema aplica a cada bloque el algoritmo SSIM para medir la calidad de cada uno.</p> <p>2.5 El sistema calcula el índice SSIM de la imagen.</p> <p>2.6 El sistema muestra una interfaz que se actualiza a medida que avanza el procesamiento de todas las imágenes del video.</p> <p>2.7 El sistema ajusta los parámetros para el cálculo del VSSIMm del video.</p> <p>2.8 El sistema finaliza el cálculo del índice VSSIMm de la secuencia de</p>

## Capítulo 3. Presentación de la solución propuesta

video, concluyendo así el caso de uso.

### Prototipo de Interfaz



### 3.5 Conclusiones

En este capítulo se han definido los conceptos asociados al modelo de dominio logrando una mejor visión del sistema y permitiendo el análisis de la solución a implementar. Se analizaron todos los requisitos de software planteados por los usuarios finales del sistema, donde se plasmó con claridad los requisitos funcionales y no funcionales. Se describieron los casos de uso del sistema, diagrama de casos de uso del sistema y actores, creando una base para sustentar las siguientes fases de desarrollo.

## **Capítulo 4. Construcción de la solución propuesta**

### **Capítulo 4. Construcción de la solución propuesta**

#### **4.1 Introducción**

El objetivo del presente capítulo es realizar el diagrama de clases del diseño, generar los principales artefactos durante los flujos de trabajo Análisis y Diseño, Implementación, Prueba y Despliegue. Se describe la construcción del sistema, las principales características de los patrones de diseño a utilizar, así como la utilización del patrón arquitectónico empleado. Se mostrará además algunos de los resultados obtenidos en las pruebas realizadas a la aplicación.

#### **4.2 Principios de diseño**

El diseño es la representación de un objeto que está siendo creado. Es una información de base que describe aspectos de este objeto. El proceso de diseño es una elaboración sucesiva de representaciones, tales como agregar información, puntos de retorno y la búsqueda alternativa. En el diseño de software se realizan una serie de procesos para definir la arquitectura, componentes, interfaces y otras características del sistema. Es además una descripción de la estructura del software que se va a implementar, los datos utilizados son parte del sistema y las interfaces entre los componentes del sistema. (33)

Los principios básicos de diseño hacen posible que el ingeniero del software transite por los procesos de diseño, estos proporcionan un marco de trabajo necesario para lograr que se realice correctamente la implementación del sistema. Favorecen la gestión de la complejidad de los sistemas software y la obtención de los factores de calidad, que estos sistemas han de presentar. (33)

##### **4.2.1 Estándares de la interfaz de aplicación**

La aplicación debe contar con una interfaz sencilla, amigable y eficiente ya que esta es un elemento clave, para que el sistema pueda cumplir con los requisitos expresados por el usuario. Las interfaces fueron diseñadas siguiendo los estándares de usabilidad expuestos en los requisitos no funcionales, con el objetivo de que el cliente pueda manipular el software sin problemas.

## Capítulo 4. Construcción de la solución propuesta

### 4.3 Definición de la arquitectura de software

La arquitectura de software se centra en los elementos estructurales del sistema, subsistemas, clases, componentes, nodos y en las colaboraciones que tienen lugar entre estos elementos a través de las interfaces. (34)

Para la construcción del sistema se utiliza el estilo arquitectónico de llamada-respuesta y dentro de este, el patrón arquitectónico en capas, específicamente en tres capas. Soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite a los implementadores la partición de un problema complejo en una secuencia de pasos incrementales. Permite optimizaciones, refinamiento, proporciona amplia reutilización y la localización de errores. El objetivo fundamental de este patrón arquitectónico es separar la lógica del negocio con la lógica de diseño y acceso a datos. Además cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. La figura 7 representa el patrón arquitectónico en tres capas, donde las clases del diseño fueron agrupadas correspondientemente en cada una de ellas: presentación, lógica del negocio y acceso a datos.

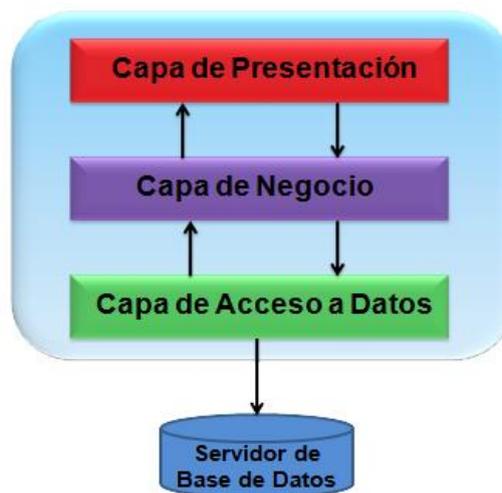


Figura 7. Arquitectura en capas. (35)

**Capa de Presentación:** en esta capa se encuentran todas las interfaces del sistema, las cuales interactúan con el usuario.

**Capa de Negocio:** esta capa posee aspectos del sistema que apoyan los procesos del negocio, permitiendo la construcción de la infraestructura de software y estableciendo las reglas que deben cumplirse. Almacena el código implementado y

## Capítulo 4. Construcción de la solución propuesta

responde a las solicitudes de los usuarios automatizando los procesos de negocio que llevan a cabo.

**Capa de Acceso a Datos:** en esta capa es donde se efectúa el enlace entre la lógica del negocio y el sistema de base de datos donde se encuentra la información a utilizar.

### 4.4 Diagrama de clases del diseño

Los diagramas de clases del diseño describen gráficamente las especificaciones de las clases de un sistema. Son utilizados para modelar de forma estática la vista del diseño y logran un mejor entendimiento a la hora de realizar la implementación. Estos diagramas poseen información sobre clases, asociaciones, atributos y métodos.

#### 4.4.1 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí para resolver un problema de diseño general en un contexto particular. (36) En la construcción del modelo de diseño fueron usados los patrones GRASP: Experto, Alta cohesión y Controlador los cuales se utilizan para asignar la responsabilidad a un objeto determinado. También fueron empleados los patrones GoF: Composición y Fachada los cuales son dirigidos al desarrollo de sistemas orientados a objetos.

- **Patrones GRASP**

**Experto:** su objetivo es asignar responsabilidades específicas a las clases que poseen la información necesaria para cumplirla. Es un principio básico que suele utilizarse en el diseño orientado a objetos. Expresa simplemente la “intuición” de que cada clase contenga los métodos relacionados con la información que posee. Además ofrece una analogía con el mundo real. (36) Un ejemplo donde se manifiesta el patrón experto es en la clase Lector\_FFmpeg (ver figura 8).



Figura 8. Diagrama donde se evidencia el patrón experto.

## Capítulo 4. Construcción de la solución propuesta

**Alta Cohesión:** este patrón es una medida de cuán relacionadas y orientadas están las responsabilidades de una clase. Planteando la contribución entre clases para realizar tareas de elevada complejidad. (36) Este patrón se evidencia en la mayoría de las clases del diseño, un ejemplo específico donde se manifiesta es en las clases VSSIMm\_Video, SSIM\_Imagen y SSIM\_Bloque (ver figura 9).

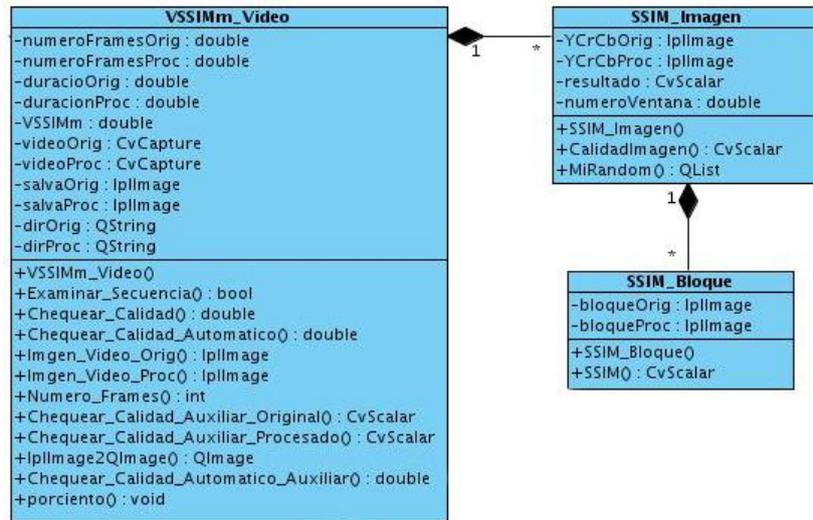


Figura 9. Diagrama donde se evidencia el patrón alta cohesión.

**Controlador:** este patrón define quién deberá encargarse de atender un evento del sistema. El controlador es un intermediario entre la interfaz de usuario y el núcleo de las clases donde se encuentra la lógica del sistema. Se debe diseñar de forma tal que no posean tanta responsabilidad, no realiza mucho trabajo por sí mismo, más bien coordina la actividad de otros objetos. (36) En la figura 10 se muestra la clase Controlador\_Calidad donde se evidencia el patrón controlador.

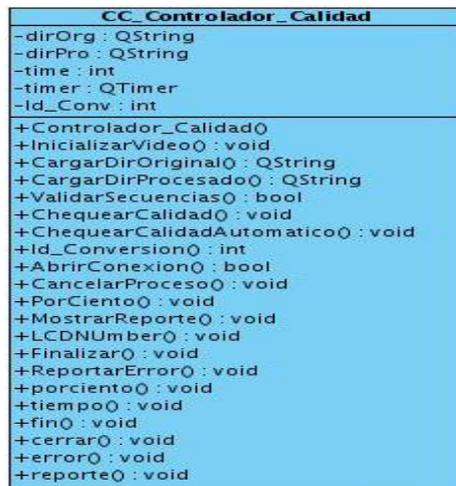


Figura 10. Diagrama donde se evidencia el patrón controlador.

## Capítulo 4. Construcción de la solución propuesta

- Patrones GoF

**Composición:** en el marco de trabajo Qt se utilizan un conjunto de patrones. Este es uno de los más evidentes que permite a los clientes tratar uniformemente a los objetos simples y compuestos de una estructura jerárquica recursiva. Es el caso del objeto QObject que define interiormente otro QObject del cual hereda. (37)

**Fachada:** brinda la posibilidad de utilizar una interfaz unificada de alto nivel. Permite ocultar toda la complejidad del sistema. Mostrando solamente al usuario puntos de entrada y el acceso a introducir valores, estando siempre ajeno al funcionamiento interno del sistema. (37) En la figura 11 se muestra la clase EvaluadorCalidad donde se pone de manifiesto el patrón fachada.

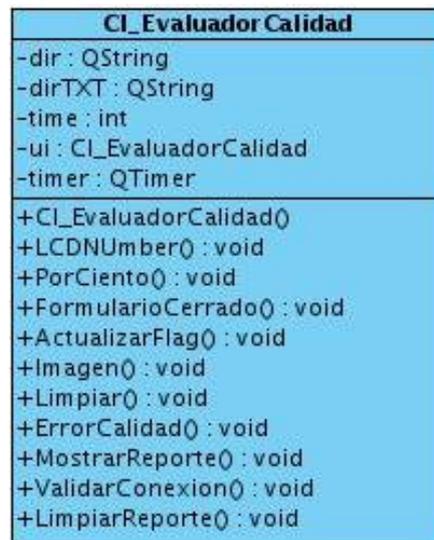


Figura 11. Diagrama donde se evidencia el patrón fachada.

### 4.4.2 Diagramas de clases del diseño por casos de uso

A continuación se muestran los diagramas de clases del diseño correspondientes a cada caso de uso del sistema.

## Capítulo 4. Construcción de la solución propuesta

### 4.4.2.1 Examinar secuencia a comparar

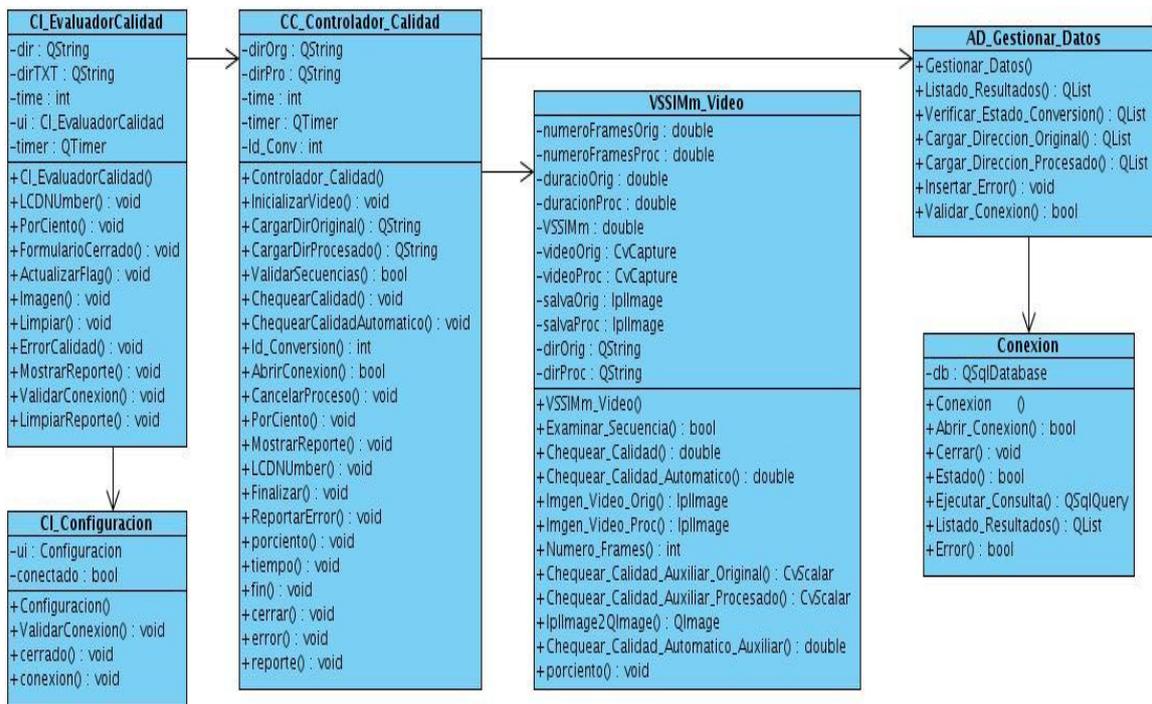


Figura 12. Diagrama de clases del diseño del CU “Examinar secuencia a comparar”.

### 4.4.2.2 Chequear calidad

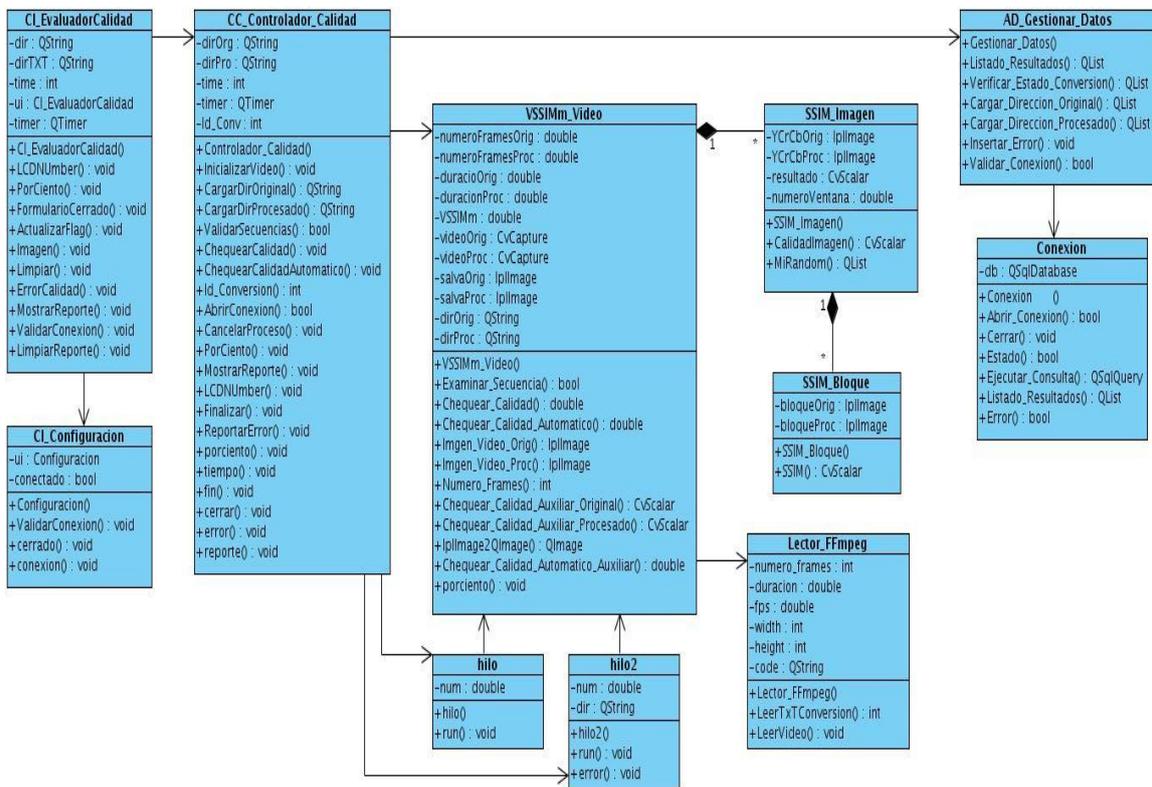


Figura 13. Diagrama de clases del diseño del CU “Chequear calidad”.

# Capítulo 4. Construcción de la solución propuesta

## 4.4.2.3 Emitir reporte de la calidad

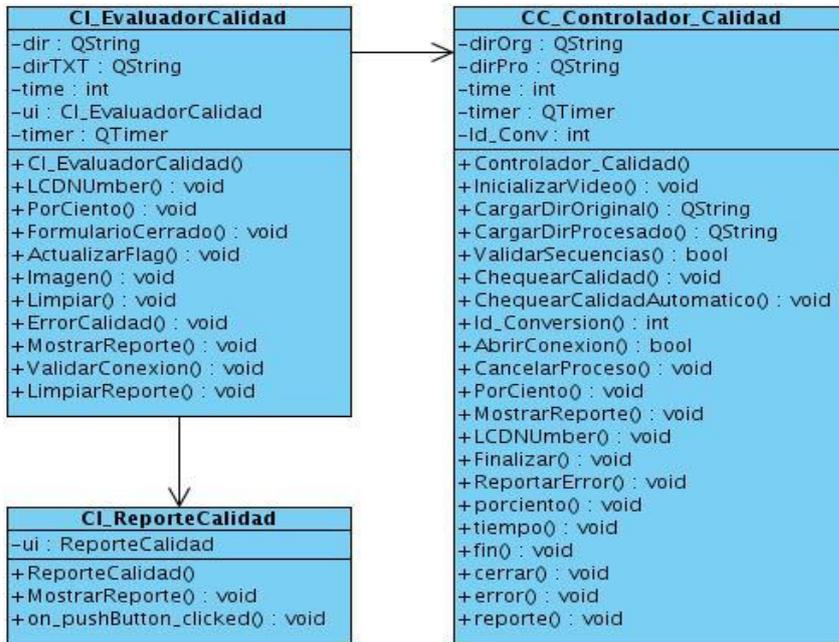


Figura 14. Diagrama de clases del diseño del CU "Emitir reporte de la calidad".

## 4.4.3 Diagrama de clases del diseño del sistema

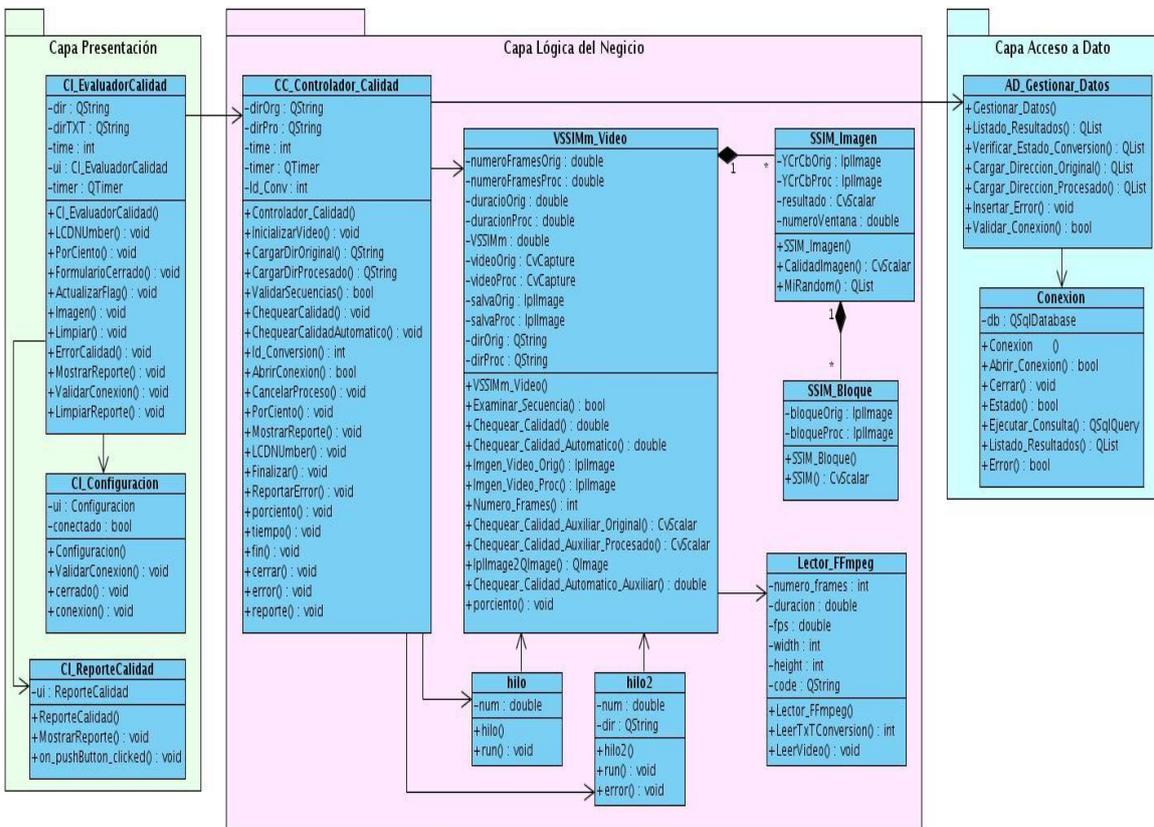


Figura 15. Diagrama de clases del diseño.

## Capítulo 4. Construcción de la solución propuesta

### 4.5 Diagrama de despliegue

El diagrama de despliegue es un grafo de nodos donde se muestran las relaciones físicas entre los componentes hardware y software en el sistema final. Modela los aspectos físicos y la vista de despliegue estática de un sistema. En la figura 16 se muestran tres nodos. El nodo **PC\_Cliente** que su función es acceder al sistema para evaluar la calidad de las medias. El nodo **Repositorio de Medias** que es donde se almacenan los videos con la calidad necesaria para ser transmitidos por la plataforma. El nodo **Servidor de Base Datos** donde se conectará la aplicación para consultar la información necesaria para realizar el proceso de evaluación de la calidad de los videos. Para establecer las conexiones al repositorio de medias utiliza el protocolo TCP/IP que permiten la transmisión de datos entre redes de computadoras y para conectarse al servidor de base datos utiliza la librería QPSQL.

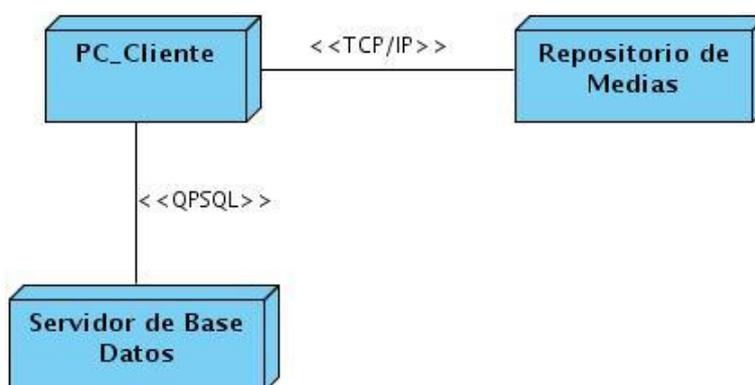


Figura 16. Modelo de despliegue del sistema.

### 4.6 Modelo de implementación

El modelo de implementación está compuesto por un conjunto de subsistemas y componentes que establece la composición física de la implementación del sistema. Para lograr una mejor organización de la implementación, se organizaron las clases por componentes lo cual permitió una mejor reutilización de código en el sistema.

#### 4.6.1 Diagrama de componente

Los diagramas de componentes son la estructura del modelo de implementación. Este diagrama presenta paquetes donde se agrupan los elementos físicos de un sistema, ya sean componentes, interfaces, así como las relaciones entre ellos (ver figura 17).

# Capítulo 4. Construcción de la solución propuesta

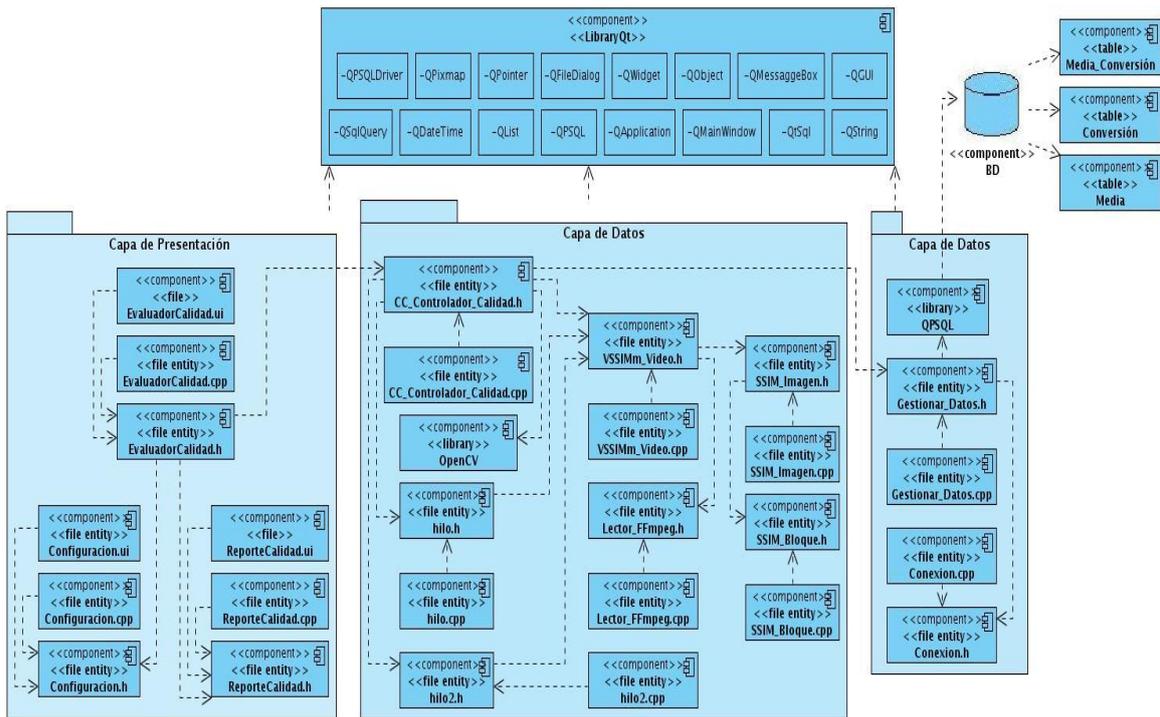


Figura 17. Diagrama de componentes del sistema.

## 4.7 Prueba del sistema propuesto

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados. Son un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. Con el objetivo de validar las funcionalidades del sistema se realizarán pruebas funcionales y unitarias para lograr buena calidad en el producto.

### 4.7.1 Pruebas funcionales

Una prueba funcional es una prueba basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. El objetivo de estas pruebas, es validar si el comportamiento observado del software cumple o no con sus especificaciones. La prueba funcional toma el punto de vista del usuario. Las funciones son probadas ingresando las entradas y examinando las salidas. (38)

Las pruebas de caja negra consisten en los requisitos funcionales de software, enfocándose directamente en la interfaz de éste, sin importar el código. Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto.

## Capítulo 4. Construcción de la solución propuesta

Los métodos de caja negra permiten derivar casos de prueba que buscan encontrar los siguientes tipos de errores: (39)

- Funciones incorrecta o faltante.
- Errores de interfaz.
- Errores en estructuras de datos o en acceso a BD externas.
- Errores de comportamiento.
- Errores de inicialización o término.

### 4.7.1.1 Casos de prueba

Un caso de prueba permite detallar la forma en que se va a probar el sistema, es un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular.

Deben verificar: si el producto satisface los requerimientos del usuario, tal y como se describe en las especificación de los requerimientos y si el producto se comporta como se desea, tal y como se describe en las especificaciones funcionales del diseño.

**Caso de prueba para el caso de uso del sistema “Examinar secuencia a comparar”.**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
<b>SC 1:</b> Escoger el tipo de evaluación.	<b>EC:</b> Escoger el tipo de evaluación.	El sistema verifica que uno de los tipos de evaluación sea seleccionado por el usuario.
<b>SC 2:</b> Examinar la secuencia original y procesada.	<b>EC 2.1:</b> Examinar la secuencia original y procesada correspondiente.	El sistema verifica que ambas secuencias fueran examinadas, listas para iniciar el proceso de validación de las propiedades imprescindibles para chequear la calidad del video, siendo estas compatibles.
	<b>EC 2.2:</b> No corresponde secuencia original y procesada.	Al examinar ambas secuencias y estas no ser compatibles después de iniciar el proceso de validación de las propiedades imprescindibles para

## Capítulo 4. Construcción de la solución propuesta

		chequear la calidad del video, el sistema muestra un mensaje de error y brinda la posibilidad de examinar nuevamente las secuencias de video.
--	--	---

Descripción de variables para el caso de uso “Examinar secuencia a comparar”.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Tipo de evaluación	radioButton	No	El usuario selecciona el tipo de evaluación que desea realizar “Evaluación Manual” o “Evaluación Automática”.
2	Video original	Campo de texto (alfanumérico)	No	El usuario examina la secuencia original a evaluar.
3	Video procesado	Campo de texto (alfanumérico)	No	El usuario examina la secuencia procesada a evaluar.

Matriz de datos para la sección “Seleccionar el tipo de evaluación”.

Escenario	Tipo de evaluación	Video Original	Video procesado	Descripción	Respuesta del sistema	Resultado de la prueba
Escoger el tipo de evaluación.	V	I	I	N/A	El sistema verifica que se haya seleccionado el tipo de evaluación que el usuario desea realizar.	Satisfactorio.

## Capítulo 4. Construcción de la solución propuesta

Matriz de datos para la sección “Examinar la secuencia original y procesada”.

Escenario	Tipo de evaluación	Video Original	Video procesado	Descripción	Respuesta del sistema	Resultado de la prueba
Examinar la secuencia original y procesada.	V	V	V	N/A	El sistema verifica que se haya examinado la secuencia original y procesada.	Satisfactorio.
No corresponde secuencia original y procesada.	V V	I V	V I	N/A	El sistema valida que las secuencias sean compatibles revisando las propiedades imprescindibles para la evaluación. De no ser compatibles muestra un mensaje de error.	Satisfactorio.

## Capítulo 4. Construcción de la solución propuesta

Caso de prueba para el caso de uso del sistema “Chequear calidad”.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
<b>SC 1:</b> Seleccionar el número de ventanas por el que desea muestrear.	<b>EC:</b> Seleccionar el número de ventanas.	El sistema verifica que quede seleccionado el número de ventanas por el que desea muestrear. Luego el sistema debe realizar los cálculos referente al VSSIMm y brindar el índice de calidad de la media procesada.

Descripción de variables para el caso de uso “Chequear calidad”.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Número de ventana	Valor numérico	No	El usuario selecciona el número de ventanas por el que desea muestrear.

Matriz de datos para la sección “Seleccionar el número de ventanas por el que desea muestrear”.

Escenario	Número de ventana	Descripción	Respuesta del sistema	Resultado de la prueba
Seleccionar el número de ventanas.	V	N/A	Después de que el sistema verifica que se haya seleccionado el número de ventana, realiza los cálculos referentes al VSSIMm brindando un índice de calidad de la media	Satisfactorio.

## Capítulo 4. Construcción de la solución propuesta

			procesada.	
--	--	--	------------	--

**Caso de prueba para el caso de uso del sistema “Emitir reporte de la calidad”.**

Caso de prueba	Emitir reporte de la calidad
<b>Condiciones</b>	Debe haberse realizado los cálculos referentes al VSSIMm.
<b>Datos de entrada</b>	Datos de la secuencia original y procesada.
<b>Resultados esperados</b>	Reporte con la información de las secuencias y los resultados obtenidos.
<b>Resultados obtenidos</b>	La prueba es satisfactoria. Se le informa al usuario el reporte de las secuencias y después de terminar la evaluación de la calidad brinda el índice VSSIMm, nivel de calidad y tiempo de ejecución.
<b>Observaciones</b>	Al realizar la prueba en el sistema se obtuvo el resultado correcto.

### 4.7.2 Pruebas unitarias

Las pruebas unitarias son una forma de probar el correcto funcionamiento de un módulo de código. Permite que los errores sean más fáciles de localizar: bastará con ejecutar la batería de “Colecciones de pruebas”, y ver qué módulos no las pasan.

Cuando se realiza prueba de la caja blanca del software se comprueba los caminos lógicos del programa, proponiendo casos de prueba que se ejerciten conjuntos específicos de condiciones y bucles. A continuación se muestran los pasos a seguir para la realización de esta técnica:

**Paso 1:** A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.

**Paso 2:** Calcular la complejidad ciclomática del grafo.

$$V(G) = NA \text{ (Número de Aristas)} - NN \text{ (Número de Nodos)} + 2$$

$$V(G) = P \text{ (Nodos predicados)} + 1$$

$$V(G) = \text{Número de regiones}$$

## Capítulo 4. Construcción de la solución propuesta

**Paso 3:** Se determina un conjunto básico de caminos independientes.

**Paso 4:** Generar los casos de prueba para la ejecución de cada camino del conjunto básico.

- Casos de pruebas de caja blanca para algunas funciones analizadas.

### Función 1. Examinar\_Secuencia.

```
//Validando las secuencias introducidas
bool VSSIMm_Video::Examinar_Secuencia()
{
    if(anchoOrig != anchoProc || largoOrig != largoProc)—————→ 1
    {
        QMessageBox* msgBox3=new QMessageBox;—————→ 2
        msgBox3->setText("Los videos no tienen la misma dimensiones");————→ 3
        msgBox3->exec();—————→ 4
        delete msgBox3;—————→ 5
        return false;—————→ 6
    }
    return true;—————→ 7
}
```

Figura 18. Fragmento de código correspondiente a la función "Examinar\_Secuencia".

- La siguiente figura muestra el grafo de flujo correspondiente al código anterior.

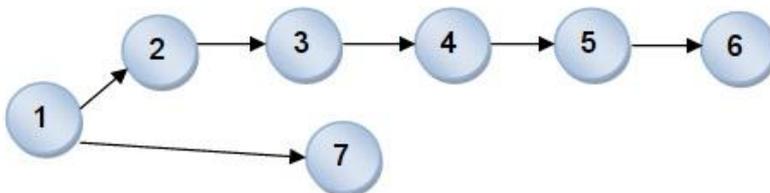


Figura 19. Grafo de flujo función "Examinar\_Secuencia".

- Calculando la complejidad ciclomática del grafo de flujo.

$$V(G) = P(\text{Nodos predicados}) + 1$$

$$V(G) = 1 + 1$$

$$V(G) = 2$$

El resultado obtenido luego de calcular la complejidad ciclomática indica un conjunto básico de caminos independientes lo cual sería la cantidad de casos de prueba que son necesarios realizar. A continuación se muestra el diseño de casos de prueba.

CB1: 1, 2, 3, 4, 5, 6

CB2: 1, 7

## Capítulo 4. Construcción de la solución propuesta

### Caso de prueba camino (1, 2, 3, 4, 5, 6)

<b>Función</b>	Examinar_Secuencia
<b>Caso de prueba</b>	1
<b>Entrada</b>	La función se invoca cuando el usuario selecciona las secuencias y oprime el botón validar.
<b>Salida</b>	La función retorna falso, ya que en este caso los videos no tienen las mismas dimensiones.

Tabla 5. Caso de prueba #1 para la función "Examinar\_Secuencia".

### Caso de prueba camino (1, 7)

<b>Función</b>	Examinar_Secuencia
<b>Caso de prueba</b>	2
<b>Entrada</b>	La función se invoca cuando el usuario selecciona las secuencias y oprime el botón validar.
<b>Salida</b>	La función retorna verdadero, ya que en este caso los videos si tienen las mismas dimensiones.

Tabla 6. Caso de prueba #2 para la función "Examinar\_Secuencia".

### Función 2. ValidarVideoEntrada.

```

bool VSSIMm_Video::ValidarVideoEntrada()
{
    if(videoOrig== NULL) → 1
    {
        QMessageBox* msgBox = new QMessageBox; → 2
        msgBox->setText("No se pudo cargar el video original"); → 3
        msgBox->exec(); → 4
        delete msgBox; → 5
        return false; → 6
    }
    if(videoProc== NULL) → 7
    {
        QMessageBox* msgBox1=new QMessageBox; → 8
        msgBox1->setText("No se pudo cargar el video procesado"); → 9
        msgBox1->exec(); → 10
        delete msgBox1; → 11
        return false; → 12
    }
    return true; → 13
}

```

## Capítulo 4. Construcción de la solución propuesta

Figura 20. Fragmento de código correspondiente al método "ValidarVideoEntrada".

- La siguiente figura muestra el grafo de flujo correspondiente al código anterior.

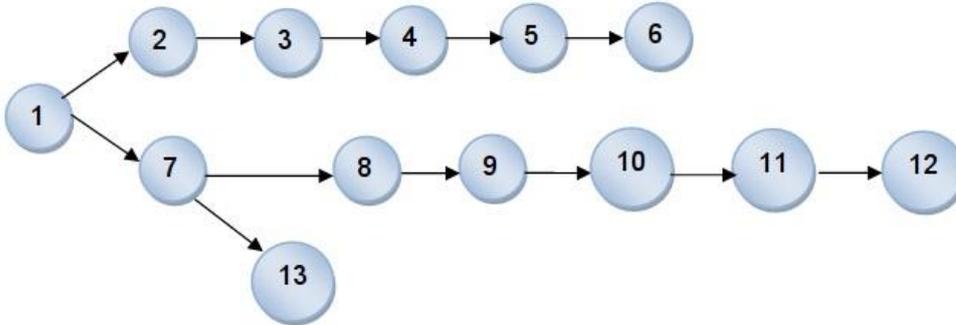


Figura 21. Grafo de flujo función "ValidarVideoEntrada".

- Calculando la complejidad ciclomática del grafo de flujo.

$$V(G) = P(\text{Nodos predicados}) + 1$$

$$V(G) = 2 + 1$$

$$V(G) = 3$$

El resultado obtenido luego de calcular la complejidad ciclomática indica un conjunto básico de caminos independientes lo cual sería la cantidad de casos de prueba que son necesarios realizar. A continuación se muestra el diseño de casos de prueba.

CB1: 1, 2, 3, 4, 5, 6

CB2: 1, 7, 8, 9, 10, 11, 12

CB3: 1, 7, 13

### Caso de prueba camino (1, 2, 3, 4, 5, 6)

<b>Función</b>	ValidarVideoEntrada
<b>Caso de prueba</b>	1
<b>Entrada</b>	La función se invoca cuando el usuario carga las secuencias de video para su posterior validación.
<b>Salida</b>	La función retorna falso, ya que en este caso la secuencia original no se pudo cargar.

Tabla 7. Caso de prueba #1 para la función "ValidarVideoEntrada".

## Capítulo 4. Construcción de la solución propuesta

### Caso de prueba camino (1, 7, 8, 9, 10, 11, 12)

<b>Función</b>	ValidarVideoEntrada
<b>Caso de prueba</b>	2
<b>Entrada</b>	La función se invoca cuando el usuario carga las secuencias de video para su posterior validación.
<b>Salida</b>	La función retorna falso, ya que en este caso la secuencia procesada no se pudo cargar.

Tabla 8. Caso de prueba #2 para la función "ValidarVideoEntrada".

### Caso de prueba camino (1, 7, 13)

<b>Función</b>	ValidarVideoEntrada
<b>Caso de prueba</b>	3
<b>Entrada</b>	La función se invoca cuando el usuario carga las secuencias de video para su posterior validación.
<b>Salida</b>	La función retorna verdadero, ya que en este caso las secuencias de video se cargaron correctamente.

Tabla 9. Caso de prueba #3 para la función "ValidarVideoEntrada".

### 4.8 Validación científica de la solución propuesta

El fin de este epígrafe es analizar los resultados obtenidos para determinar si se cumplen los objetivos marcados y evaluar la eficacia del sistema propuesto, comprobando si los resultados del algoritmo VSSIMm se ajustan a la opinión subjetiva.

Para validar la solución propuesta se realizaron pruebas a un total de 48 videos pertenecientes a la base de datos del Laboratorio de Ingeniería para Imagen y Video de la Universidad de Texas (LIVE según sus siglas en inglés). Para el proceso se utilizaron tres videos sin comprimir, de alta calidad, mayormente basados en escenarios naturales. Todos estos videos se filmaron con un equipo profesional de gama alta y se convirtieron en formato digital con el mayor cuidado, garantizando que los videos no fueran distorsionados. (40)

## Capítulo 4. Construcción de la solución propuesta



(a) Pedestrian Area

(b) Blue Sky

(c) Mobile & Calendar

Figura 22. Representa los videos originales obtenidos de la base de datos LIVE. (40)

Por cada video sin comprimir se crearon 16 secuencias de pruebas aplicando cuatro procesos diferentes de distorsión: compresión MPEG-2, compresión H264, errores de transmisión a través de redes IP y errores de transmisión a través de redes inalámbricas; aplicándole a un video cuatro veces la misma distorsión para obtener niveles diferentes de calidad. El rango de valores que fueron concebidos en la base de datos LIVE para emitir esta clasificación según la métrica VSSIMm fue el siguiente: Excelente (1 / 0.95), Bueno (0.95 / 0.91), Regular (0.91 / 0.86) y Malo (0.86 / 0). A continuación se muestran datos estadísticos que ejemplifican lo expresado. (40)

Secuencia de evaluación	VSSIMm	Nivel de Calidad
Pa1_25fps.yuv	0.97938	Excelente
Pa2_25fps.yuv	0.94882	Buena
Pa3_25fps.yuv	0.89937	Regular
Pa4_25fps.yuv	0.84753	Mala
Bs1_25fps.yuv	0.98341	Excelente
Bs2_25fps.yuv	0.94719	Buena
Bs3_25fps.yuv	0.88230	Regular
Bs4_25fps.yuv	0.83529	Mala
Mc1_25fps.yuv	0.98935	Excelente
Mc2_25fps.yuv	0.94874	Buena
Mc3_25fps.yuv	0.90632	Regular
Mc4_25fps.yuv	0.82586	Mala

Tabla 10. Representa el índice y nivel de calidad utilizando VSSIMm para las secuencias analizadas.

## Capítulo 4. Construcción de la solución propuesta

Para darle mayor validez y robustez a las pruebas realizadas se utilizó la herramienta MSU Quality Measurement Tool, apoyándose en el algoritmo SSIM (precise). Su elección se debe al grado de similitud existente entre ambas métricas debido a que éste último es la base de VSSIMm. (17)

Las secuencias empleadas para las pruebas experimentales fueron tomadas teniendo en cuenta los formatos estipulados por la plataforma PTARTV (h264 y ogg) para la transmisión de las medias. Además se valoró la compatibilidad con otros tipos de codec, considerando posibles cambios en los formatos de transmisión.

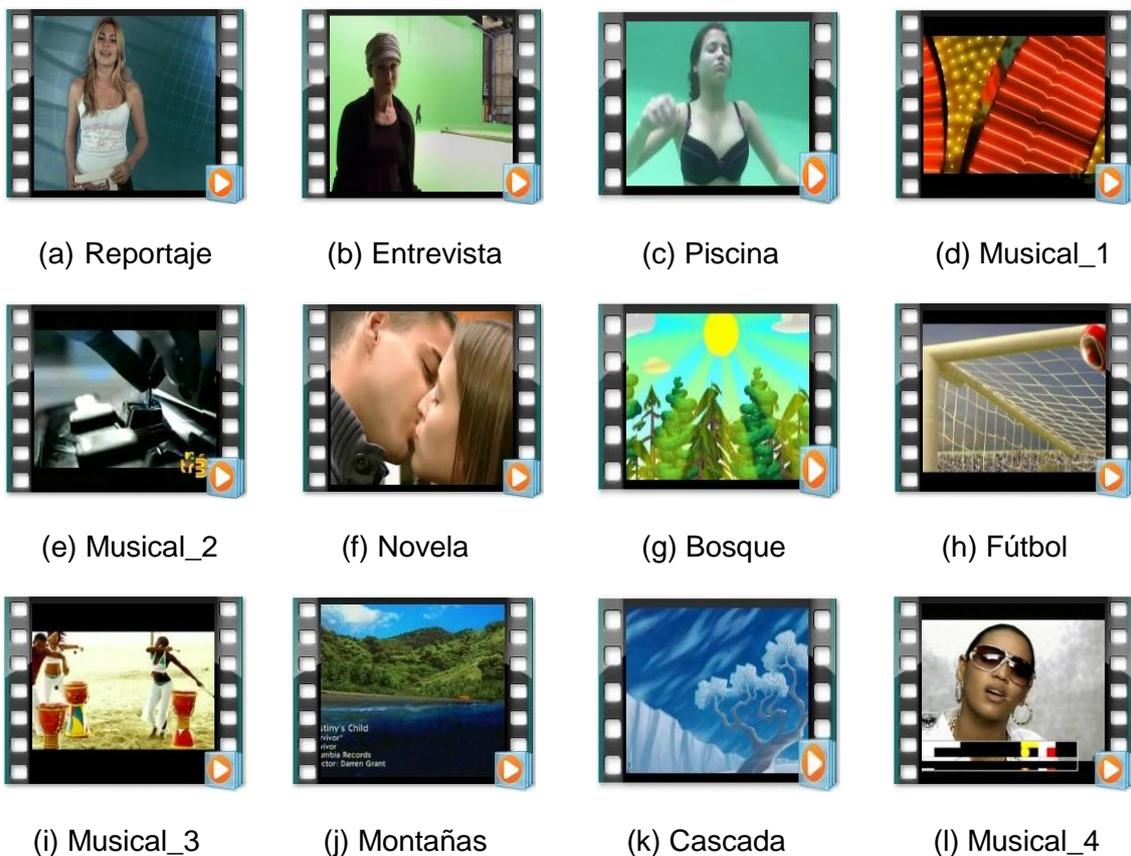


Figura 23. Representa las secuencias de videos utilizadas para la validación de la métrica implementada.

A continuación se expone una tabla con los resultados arrojados por las pruebas experimentales.

Secuencia de evaluación	SSIM (precise)	VSSIMm
Reportaje (h264)	0.96719	0.961775
Reportaje (ogg)	0.96512	0.964395

## Capítulo 4. Construcción de la solución propuesta

Reportaje (avi)	0.86463	0.836181
Entrevista (h264)	0.95836	0.954972
Entrevista (ogg)	0.96439	0.948993
Piscina (mpeg2)	0.98829	0.983785
Piscina (mpeg4)	0.80723	0.785248
Musical_1(avi)	0.96116	0.934819
Musical_2 (avi)	0.95917	0.915729
Novela (mpeg4)	0.81520	0.786247
Bosque (h264)	0.97865	0.958932
Futbol (ogg)	0.94437	0.921125
Musical_3 (mpeg2)	0.90829	0.878297
Montañas (avi)	0.93876	0.926794
Cascada (h264)	0.82385	0.792533
Musical_4 (avi)	0.84463	0.816579

Tabla 11. Estimaciones de los algoritmos para las secuencias de evaluación.

La métrica implementada ofrece un tiempo de ejecución realmente bajo, con buenos resultados respecto al algoritmo SSIM (precise). La métrica VSSIMm para la secuencia Entrevista (ogg) de 40 segundos de duración brinda como resultados un índice de 0.948993 y tiempo de ejecución de 21 segundos. Sin embargo para esta misma secuencia el algoritmo SSIM (precise) provee un índice de 0.96439 para un tiempo de ejecución de 56 segundos. Debe tenerse en cuenta que la diferencia, en cuanto a valores obtenidos, es producto a que la métrica implementada incluye análisis de movimiento y actividad en la secuencia, parámetros que no son tenidos en cuenta para el cálculo del SSIM (precise) y afectan el resultado final.

## **Capítulo 4. Construcción de la solución propuesta**

### **4.9 Conclusiones**

En este capítulo se definieron los diagramas de clases de diseño para cada caso de uso, así como el diagrama de diseño general del sistema. Lo que permitió el inicio de la implementación de la aplicación. Se estableció el patrón arquitectónico en tres capas para lograr un mejor funcionamiento del sistema, así como los patrones de diseño más factibles para la construcción del mismo. Quedaron descritos los principios de diseño para el sistema propuesto, definiendo además el diagrama de despliegue y el diagrama de componentes. Finalmente se realizaron pruebas para validar la solución propuesta, las cuales arrojaron como resultado que la aplicación cumple de manera satisfactoria, con las funcionalidades identificadas durante el proceso de desarrollo del software llevado a cabo como parte de la presente investigación.

## Conclusiones generales

### Conclusiones generales

Video Quality Tool es el nombre que recibe el sistema desarrollado para evaluar la calidad de los videos convertidos en la plataforma PTARTV basado en métricas de distorsión estructural. Una vez realizadas todas las tareas de investigación científica, es posible afirmar que se han cumplido satisfactoriamente, todos los objetivos propuestos a lo largo del presente trabajo de diploma arribando a las siguientes conclusiones:

- Las aplicaciones informáticas dirigidas a la evaluación de calidad en secuencias de video, son desarrolladas principalmente por países del primer mundo siendo muy pocas las soluciones que existen en este campo, pagando altos precios los países del tercer mundo por el uso de esta tecnología. Además el análisis de las herramientas existentes arrojó como resultado que no son adaptables ni configurables al flujo de trabajo de la plataforma.
- La metodología de desarrollo, lenguajes utilizados, herramientas y biblioteca incluida ofrecieron el soporte necesario para lograr un producto con los requerimientos deseados. Además de brindarle al mismo una calidad y rendimiento acordes con las exigencias planteadas por el usuario.
- La implementación del algoritmo VSSIMm contribuyó a elevar el nivel de calidad de las transmisiones televisivas, ya que se logró desarrollar el producto final que evalúa calidad de los videos en la plataforma PTARTV.
- El trabajo realizado tributa a la plataforma PTARTV como experiencia en el procesamiento de imagen y video digital.
- La solución implementada agiliza y automatiza el chequeo de la calidad en la plataforma PTARTV.
- La utilización de la solución informática VQT en una televisora, posibilita disminuir el número de recursos humanos y materiales para el chequeo de la calidad en los videos.
- La presente investigación representa para el país un avance científico en cuanto a evaluación de calidad en videos se refiere, debido a que hasta entonces esta temática no ha sido tratada por documentos investigativos. Teniendo en cuenta además que institutos como el Departamento de Comercio de los Estados Unidos, la Universidad de Carlos III de Madrid y Universidad Estatal de Moscú pertenecientes a países desarrollados, son los precursores en esta área de investigación.

### Recomendaciones

Es válido destacar que durante el progreso de la investigación se han identificado algunas mejoras, que podrían implementarse en un futuro para darle una mayor exactitud al producto obtenido. Dichas recomendaciones son:

- Lograr la integración de las herramientas Video Quality Tool y Digital Video Quality siendo ambas desarrolladas por el Departamento de Señales Digitales de la Facultad 6, de la Universidad de las Ciencias Informáticas. Una basada en métricas de distorsión estructural y otra basada en métricas de detección de errores conformarían un perfecto modelo híbrido para el chequeo de la calidad general en videos digitales.
- Investigar y analizar otras métricas de evaluación de la calidad de los videos como VQM y VESSIM con el objetivo de integrarlas al sistema propuesto para optimizar el proceso de chequeo de calidad.
- Debido a los constantes cambios a los que está sujeto los métodos de evaluación objetiva de la calidad, actualmente es objeto de estudio en importantes centros de investigación, se recomienda mantener un seguimiento de las posibles actualizaciones en la métrica VSSIMm en aras de mejorar el rendimiento y obtener resultados más confiables.

## Referencias bibliográficas

### Referencias bibliográficas

1. **Campos, Armando Mora.** Estudio de Arquitecturas VLSI de la Etapa de Predicción de la Compensación de Movimiento, para Compresión de Imágenes y Video con Algoritmos Full-Search. Aplicación al Estándar H.264/AVC. Valencia : Universidad Politécnica de Valencia, 2008.
2. **Fraga, Luis Gerardo de la.** Procesamiento Digital de Imágenes. 7 de Mayo de 2001.
3. **Manuel Agustí, Jose V. Benlloch y Vicente Atienza.** Adquisición y representación de medios digitales. Curso 2009\_2010.
4. **Universidad Don Bosco, Facultad: Humanidades, Producción Avanzada de Video. Guía 4.** El Video Web, La Compresión de Video yadobe media Encoder. El Salvador : s.n., 2010.
5. **González, D. José A. Verdoy.** SVD para la transmisión progresiva de imágenes y la codificación de video digital. Valencia : s.n., 2009.
6. **Sainz, Javier Herrán.** Diseño e implementación FPGA de un Estimador de Movimiento de Tamaño Variable para el H.264. Cantabria : s.n., Septiembre 2007.
7. **Hormigos, Carlos Esteban Baz.** Archivo Abierto Institucional de la Universidad Carlos III de Madrid. [En línea] 2009. [http://e-archivo.uc3m.es/bitstream/10016/7202/2/PFC\\_Carlos\\_Esteban\\_Baz\\_Hormigos.pdf](http://e-archivo.uc3m.es/bitstream/10016/7202/2/PFC_Carlos_Esteban_Baz_Hormigos.pdf).
8. **Symes, P.** Video Compression Demystified. USA : McGraw-Hill, 2001.
9. **González, Rafael C. y Goods, Richard E.** Digital Image Procesing. New Jersey, USA : Upper Saddle River, 1992.
10. **Bimbo, Alberto de.** Bases de Datos de Imágenes y Videos. Ciudad de la Habana : s.n., 2005.
11. **Joskowicz, José.** Estudio de la Medida de la Calidad Perceptual de Video. 2008.
12. **Televisión, Grupo Técnico del Foro de la.** Televisión Digital Terrestre. [En línea] [Citado el: 20 de Noviembre de 2010.]

## Referencias bibliográficas

<http://www.televisiondigital.es/TecnologiasRelacionadas/AltaDefinicion/ForoTVAD/ConclusionesForo/8TH2CALIDADDEVIDEOENTVDEALTADEFINICI%C3%93N.pdf>.

13. **Poirier, F. y otros.** TAPAGE : édition de tableaux sur ordinateur à stylo vers une désignation naturelle. Lyon : s.n., 1993.

14. **Wang, Zhou; Lu, Ligang; Bovik, Alan C.** The Center for Neural Science at New York University. [En línea] 2002.  
<http://www.cns.nyu.edu/~zwang/files/papers/icassp02a.pdf>.

15. **Wang, Zhou, Lu, Ligang y Bovik, Alan C.** Laboratory for Image & Video Engineering. [En línea] 2004.  
[http://live.ece.utexas.edu/publications/2002/zw\\_icip2002\\_vidqual.pdf](http://live.ece.utexas.edu/publications/2002/zw_icip2002_vidqual.pdf).

16. **Shengnan Ye, Kaina Su, Chuangbai Xiao.** Video Quality Assessment Based on Edge Structural Similarity. Beijing, China : Universidad Tecnológica de Beijing. 2008.

17. **Dr. D. Vatolin, el Dr. M. Smirnov, Ratushnyak A., V. Yoockin.** Compression Project. [En línea] 2001. [Citado el: 3 de Diciembre de 2010.]  
[http://www.compression.ru/video/quality\\_measure/info\\_en.html](http://www.compression.ru/video/quality_measure/info_en.html).

18. **Evans, Donald L.** Video Quality Measurement Techniques. s.l. : U.S. Department of Commerce. 2002.

19. **Chacón, Julio César Rueda.** Aplicación de la Metodología RUP para el desarrollo rápido de Aplicaciones Basado en el Estándar J2EE. Guatemala : s.n. Marzo de 2006.

20. **Q, Jose M. Batista.** Programación Extrema (XP). Bolivia : s.n.

21. Ayuda de RUP. s.l. : Corporación IBM, 2006.

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

23. **González, Carlos Daniel.** Usabilidad web. Introducción a C++ y a la Resolución de problemas. [En línea] 2008. [Citado el: 22 de 01 de 2011.]  
<http://www.usabilidadweb.com.ar/cpp.php>.

24. **Universidad, Técnica Federico Santa María. Departamento de Informática.** Introducción a Java. 2003.

## Referencias bibliográficas

25. **Rocha, Jorge Luis Aréchiga.** 50MEGS. [En línea] [Citado el: 14 de Marzo de 2011.] <http://arechiga.50megs.com/tpoo2/javah1.html>.
26. **Álvarez, Sara.** desarrolloweb.com. [En línea] [Citado el: 18 de Febrero de 2011.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
27. **TLDP, Equipo de desarrollo de.** Manual de Usuario de PostgreSQL. [En línea] 2010. [Citado el: 25 de Febrero de 2011.] <http://es.tldp.org/Postgresql-es/web/navegable/user/user.html>.
28. **Nokia.** QT. [En línea] 17 de Diciembre de 2008. [Citado el: 06 de Junio de 2010.] <http://qt.nokia.com/>.
29. The Open Computer Vision Library. [En línea] 2003. <http://www.intel.com/software/products/opensource/libraries/cvfl.htm>.
30. **Carranza Athó Fredy, Florian Cruz Laura.** OPENGL y VTK. Trujillo Perú : s.n., 2006.
31. **Garcerant, Iván.** Tecnología y Synergix. [En línea] 2008. [Citado el: 17 de Febrero de 2011.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
32. **Chaves, Michael Arias.** La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. Costa Rica : s.n. 2006.
33. **Peñalvo, Dr. Francisco José García.** Principios del diseño del software. 2008 .
34. **Computación, Escuela de Ingeniería de Sistemas y.** Arquitectura. Universidad del valle : s.n.
35. **Llorente, César de la Torre, y otros.** Guía de Arquitectura N-Capas orientada al Dominio con .NET 4.0. 2010.
36. **Larman, Craig.** UML y Patrones. Montevideo Uruguay : s.n. 1999.
37. **Madrid, Facultad de informática - Universidad Politécnica de.** Patrones del "Gang of Four" . Madrid : s.n.
38. **Lamanca, Beatriz Pérez.** Gestión de las Pruebas Funcionales. Universidad de la República, Montevideo, Uruguay : s.n., 2007.
39. **Tello, Dr. Eduardo A. Rodríguez.** Importancia de las pruebas de software. 2011.

## Referencias bibliográficas

40. **Bovik, Alan C., y otros.** Study of Subjective and Objective Quality Assessment of Video. 2009.
41. **M. Martinez-Rach, y otros.** PSNR vs. quality assessment metrics for image and video codec. s.l. : Dept. de Informática de Sistemas y Computación Universidad Politécnica de Valencia.
42. **Vargas, Manuel.** Metodología de desarrollo de la herramienta de captura de información de especies y especímenes de la REyE de IABIN. Costa Rica : s.n., Febrero de 2008.
43. **Santos., Carelys Suárez Arencibia & Anibal Santos.** SIGNUC: Sistema para la gestión de la información de muestras de núcleos en el CEINPET. Ciudad de La Habana : s.n., 2009.
44. **Flanagan, Davis.** Java en pocas palabras. México : s.n. 1999.
45. **Mora, Sergio Luján.** C++ paso a paso.
46. **España, Grupo Técnico del Foro de la Televisión de Alta Definición en.** Calidad de Video en la Alta Definición. Madrid : s.n. Marzo de 2008.
47. **Lopes, Karine Petry y Paula marien Albrecht.** Modelos para Interoperabilidades de Sistemas Hospitalares Utilizando Padrao HL7. Florianópolis : s.n., 2005.
48. **Sánchez, Esmeralda Guindel.** Calidad y Seguridad de la Información y Auditoría Informática. Madrid España : s.n., Noviembre de 2009.
49. **Sergio De La Llana Alamar, Samuel Molina Vinci, Sergio Sanchez Martinez.** Introducción a las Librerías OPENCV. [En línea] [Citado el: 06 de Diciembre de 2010.] <http://web-sisop.disca.upv.es/~imd/cursosAnteriors/2k3-2k4/copiaTreballs/serdelal/trabajoIMD.xml>.
50. **Simoncelli, Zhou Wang and Eero P.** An Adapttime Linear System Framework for Image Distortion Analysis. New York : s.n.
51. **Solís., Manuel Calero.** Una explicación de la programación extrema (XP). Madrid : s.n. 2003.
52. **Wang, Zhou, Member, Student y Bovik, Alan C.** A Universal Image Quality Index. s.l. : IEEE Signal Processing Letters. 2002.

## Referencias bibliográficas

53. **B, Ing. Alexander Oré.** CalidadSoftware.com. [En línea] 2009. [Citado el: 17 de Mayo de 2011.] [http://www.calidadyssoftware.com/testing/pruebas\\_funcionales.php](http://www.calidadyssoftware.com/testing/pruebas_funcionales.php).
54. **Blank, Isabel, Herrera, Larissa y Ortiz, Miguel.** Pruebas de Funcionalidad. 2005.
55. **Dábalos, Gabriel.** El apagón de la televisión analógica en Cuba: 10 respuestas sobre la televisión digital. [ed.] La Habana Cuba: s.n. 2010.
56. **Berenguer, Abel Días y Román Vieito, Alberto Ramón.** Sistema de Administración y Configuración de la solución de Captura y Catalogación de Medias. Ciudad de la Habana : s.n., 2009.
57. **Ivar Jacobson, Grady Booch y James Rumbaugh.** El Lenguaje Unificado de Modelado.Manual de Referencia. 2000.

### Glosario de términos

#### “A”

**Actores:** toda entidad externa al sistema o negocio, que guarda una relación con este y que demanda el uso de alguna de sus funcionalidades.

**Aplicación:** es una clase de programa informático creado para facilitar al usuario un determinado tipo de trabajo. Esto lo caracteriza frente a otros programas como los sistemas operativos, las utilidades y los lenguajes de programación.

#### “C”

**Clase:** una clase representa un conjunto de objetos que tienen una estructura, un comportamiento y unas relaciones con propiedades parecidas.

**Casos de uso:** es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema.

#### “G”

**GoF:** acrónimo de los vocablos en inglés: Gang of Four, que en español equivalen: a Grupo de los Cuatro o Banda de los Cuatro. Este es el nombre con el que se conoce comúnmente a los autores del libro Design Patterns, referencia en el campo del diseño orientado a objetos, estos son: Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides.

#### “H”

**Herramientas CASE:** la herramienta CASE (Computer Aided Systems Engineering) cuyo significado en español es ingeniería de sistemas asistida por ordenador, es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo de sistemas. Donde se pueden realizar modelos y diagramas, diseño de proyectos, cálculo de costes, documentación o detección de errores entre otras.

#### “I”

**IDE:** es el acrónimo de los vocablos en inglés: Integrated Development Environment, que en español sería: Entorno de Desarrollo Integrado. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir,

## Referencias bibliográficas

consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

### “M”

**Media:** video, imagen o cualquier otro material audiovisual que requiere el uso especial de un equipamiento para su visualización.

### “S”

**Sistema:** Conjunto de elementos interrelacionados entre sí que tienen un propósito determinado. En términos de gestión, cuenta con varios procesos básicos: las entradas, el procesamiento, el almacenamiento y las salidas.