Universidad de las Ciencias Informáticas
Facultad 6

Título: Evaluación de la calidad de los videos en la plataforma PIARTV basado en métricas de detección de errores.

Trabajo de Diploma para Optar por el Título De Ingeniero en Ciencias Informáticas

Autores:

Magalys Yisel Menéndez Verde

Ana Lizandra Aleman Martínez

Tutor:

Ing. Dunier Domínguez Mora

Cotutor:

Ing. Rocny Morales Delgado

La Habana, junio del 2011.

"Año 53 del Triunfo de la Revolución"

Declaración de autoría

Declaramos que somos los únicos autores del trabajo titulado:

Evaluación de la calidad de los videos en la plataforma PTARTV basado en métricas de detección de errores.

Autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

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

_____ _____

Ana Lizandra Aleman Martínez **Magalys Yisel Menéndez Verde**

_____

Ing. Dunier Domínguez Mora

Agradecimientos

Ana Lizandra:

Agradecerle a la Revolución y a la Universidad de las Ciencias Informática por darme esta oportunidad de poder formarme como una profesional.

A mis padres Sofía y Agustín por ser los mejores padres del mundo, que tanto esfuerzo y empeño han puesto para que yo pueda haber hecho este sueño realidad. Por ser mis dos fuentes de inspiración en la vida y mis paradigmas a seguir. Por ser los primeros maestros y hacer de mí la persona que soy, por enseñarme a vivir, por sus consejos y toda la confianza que siempre han depositado en mí. Por el amor infinito y el cariño que siempre me han dado. Porque me debo incondicionalmente a ustedes y se merecen todo mi esfuerzo y voluntad. Los quiero mucho.

A mi esposo Robert A. por ser tan bueno conmigo y soportarme todas mis malcriadeces.Por apoyarme en todo momento y por amarme tanto.Gracias por ser lo mejor que me ha pasado esta vida. Te amo mucho.

A mis suegros Mireisy y Alberto por ser mis segundos padres,por dejarme ser su niña. Por el amor infinito y el cariño que siempre me han dado y apoyarme en todos los momentos difíciles por los que pasé.

A Yamy y Kelvis por malcriarme tanto.Por brindarme tanto amor y cariño.Hoy no están cerca de mi ,pero sé que siempre me aconsejarán y apoyarán desde donde estén.Gracias ,siempre los amaré.

A nuestro tutor Dunier. Por su preocupación y ayuda que nos proporcionó en la elaboración de este trabajo de diploma.

A nuestro co-tutor Rocny. Por dedicar su tiempo en ayudarnos ,aconsejarnos y rectificarnos.Gracias por todo.

A mi familia en general por apoyarme anímica y moralmente durante todos estos años.

A la familia de mi esposo por su preocupación, por mi tesis y su apoyo incondicional hacia mi.Gracias los quiero.

A mis hermanos que de una manera u otra me apoyaron, y se que están orgullosos de mi.

A todas mis amistades por apoyarme en todo momento y depositar su confianza en mi especialmente: Yamir, Yamila, Alejandro, Ana Lilian, Mónica, Yiyi, Yeslidier, Lisett, Lida, Miguel, Pedro, Handy y Yailenys.

Al tribunal por todas sus sugerencias.

A mi compañera de tesis Magalys por todo lo que aprendimos y compartimos juntas.Por luchar juntas para sacar adelante la tesis, que tanto trabajo nos dió.

Y en general a todas aquellas personas que estuvieron presentes en cada paso que di para convertirme en la persona que soy.

Magalys Yisel:

A mis padres Miriam y Hugo, por su amor y cariño incondicional, porque nadie ha confiado en mi como ellos, Por hacerme la persona que hoy soy, gracias a su apoyo y dedicación, porque siempre han sido mi ejemplo a seguir y las personas que más admiro y de las cuales me siento profundamente orgullosa. Porque son toda mi vida, porque sin ustedes no sería nada. Los quiero con todo mi corazón,

A mi novio Rocny por ser el amor de mi vida, por apoyarme y ayudarme como nadie en la realización de la tesis, por hacer que estos últimos cuatro años fuesen los más felices de mi vida y sobre todo por quererme y darme tanto cariño y amor.Te Amo.

A mi hermano que aunque es el más gruñon de la familia siempre está pendiente de mis cosas,a mi cuñada Lisleidy y a la luz de mi familia, mi sobrino,

A mi tío Pupi, a mi abuelita linda y a mi primo Luisito por siempre confiar en mi y quererme mucho.

A mi abuelo Cuco por su cariño y porque sé que soy su nieta favorita(la única).

A mi abuelo Chicho que aunque no pudo estar aquí conmigo compartiendo mi felicidad y riendose como siempre estaba, sé que donde esté se siente muy orgulloso de todo lo que he logrado.

A mis amistades Maylin, Yelen y Jorgito que han estado en todo momento a mi lado, algunas veces riendonos y otras no tan contentos, pero siempre juntitos.

A Dunier por toda su ayuda y confianza.

A todos mis amigos y compañeros de estudio que han hecho que estos cinco años sean más fáciles y placenteros. A Yudalys, Maria Luisa, Yenia, Elizabeth y muchísimos más.

A mi compañera de tesis, por luchar juntas para sacar adelante la tesis, que tanto trabajo nos dió.

A todos los que han puesto una granito de arena para la realización de esta tesis, en especial a Pedrito que nos ayudó muchísimo.

A los miembros del tribunal por todas sus sugerencias.

A todos muchas gracias.

Dedicatoria

Dedico este trabajo de diploma a las personas más importantes en mi vida que han puesto toda su fe en mi y han hecho todo lo posible por que pueda optar por el título de Ingeniera en Ciencias Informática:

A mi madre Sofía Martínez Bravo.

A mi padre Agustín Aleman Gutiérrez.

Al amor de mi vida, mi esposo Robert A. Peña Yantá.

A mis hermanos Mario A. y Dennis.

A mis segundos padres y suegros Mireisy y Alberto.

A mi cuñada y hermanita Yamilys y Kelvis.

A mi familia en general.

A todos mis amigos por apoyarme en todo momento y confiar en mí, gracias por compartir lo mejor de sí mismos y por aceptarme en sus vidas.

A todos los quiero.

Ana Lizandra.

A mi mamá y mi papá, por ser mi guía y apoyo en todo momento.

A mi novio por brindarme todo su amor.

A mi hermano y mi sobrino.

A mi abuela Mirtha.

A mi tío Pupi.

A mi primo Luisito.

A mis abuelos Chicho y Cuco.

A mis amistades de los 5 años de carrera, Maylin, Yelen y Jorgito.

A todos los que de una forma u otra constribuyeron enla realización de esta tesis.

A todos los quiero mucho.

Magalys.

Resumen

Los medios de comunicación son de extrema importancia para el desarrollo, entretenimiento y educación de la sociedad, en mayor o menor grado según el nivel de desarrollo que se tenga. Es así como la televisión en los últimos años se ha convertido en un importante medio de difusión masiva. En un principio los sistemas de grabación de video eran analógicos, más adelante con el desarrollo de las tecnologías surge el video digital, que inicialmente era de baja calidad, pero para el mercado de consumo tomó un gran auge en un muy poco tiempo, alcanzando un desarrollo notable.

En muchas ocasiones los videos son sometidos a un proceso de conversión con el objetivo de tenerlos almacenados con un mismo formato. En ocasiones este proceso da lugar a errores en la conversión, lo cual hace que los mismos no alcancen la calidad requerida para ser transmitidos. Esta situación provoca que se consuma más tiempo de lo previsto, o incluso que en algunas circunstancias sean transmitidos con estos errores, lo que puede causar molestias innecesarias a los televidentes.

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. Dentro de esta última clasificación se agrupan las métricas basadas en detección de errores, las cuales se enfocan en detectar las diferentes errores que se producen en la secuencia de video convertida con relación a la original.

El objetivo de este trabajo de diploma es desarrollar un sistema que permita automatizar la evaluación de la calidad en los videos de la plataforma de Transmisión Abierta para Radio y Televisión (PTARTV), basada en métricas de detección de errores. Además en este documento se recoge toda la investigación realizada para la elaboración del sistema, con la correspondiente explicación de los conceptos y características más trascendentales de este. Se exponen igualmente los resultados del proceso de desarrollo del software.

Palabras Claves: Videos, Proceso de Conversión, Calidad, Métrica, Errores.

Summary

The media are extremely important for development, entertainment and education of society to a greater or lesser degree depending on the level of development you have. This is how the television in the last few years of this century has become an important means of mass media. At the begining the video recording systems were analog, later with the development of comunications and internet the digital video technology emerges, which initially was low quality, then for the world market took a big boom in a very short time, reaching a very important development.

In many cases the videos are subject to a conversion process in order to have them stored in the same format. Sometimes this process leads to errors in the conversion which makes them to not meet the required quality to be transmitted. This situation causes them to consume more time than expected, or even in some circunstances to be transmitted with these errors, which causes a lot of inconveniences to spectators.

In the world of visual processing have been established metrics to assess the quality of video sequences. These metrics fall into two main groups: subjective and objective. Within the last one classification are grouped the based metric error detection, which focus on identifying the various errors that occur in the converted video stream with respect to the original.

The objective of this dissertation is to develop a system to automate the evaluation of the quality of the videos of the Open Broadcasting platform for radio and television (PTARTV) metrics based on error detection. Additionally, this document contains all the research done to develop the system, with accompanying explanation of the concepts and features of this far-reaching. Also discussed are the results of the software development process.

Keywords: Video, Conversion Process, Quality, Metrics, Bugs.

Índice de Tablas

Índice de Figuras

Índice de Contenido

Introducción

El largo camino de la evolución del hombre ha estado poblado por diversas dificultades, entre las cuales se encuentran la necesidad de comunicarse y mantenerse informado. Esto hizo que se prestara especial atención al desarrollo de los medios de comunicación. El surgimiento de la televisión constituyó un gran salto, convirtiéndose en uno de los medios más utilizado a nivel mundial.

La televisión ha estado en constante desarrollo y unos de los aspectos fundamentales en dicha evolución han sido los ordenadores. Los cuales desde su creación han estado en un constante y vertiginoso avance hasta hacer posible la digitalización de la información. Este es el mecanismo más sencillo, eficiente y rentable para almacenar, administrar y consultar información. En sus inicios solo se digitalizaban ficheros que contenían poca información debido a que los dispositivos de almacenamiento no contaban con la capacidad necesaria. Esto constituyó una limitación en el desarrollo de la informática y las comunicaciones. Haciendo que fuera indispensable la construcción de nuevos dispositivos de almacenamientos que permitieran la digitalización de archivos más grandes como imágenes y videos.

El video digital es un tipo de sistema de grabación que funciona usando una representación digital de la señal de video. La tecnología digital se introdujo por primera vez en 1983 con el formato D-1 de Sony, que grababa una señal no comprimida de video componente[1] de definición estándar en forma digital en vez de las formas analógicas que habían sido frecuentes hasta ese momento. El video digital para el mercado de consumo apareció por primera vez en la forma de QuickTime, la arquitectura de Apple Computer para los formatos de datos basados en tiempo y streaming[2]. Aunque al principio era de baja calidad, el video digital para el mercado de consumo mejoró rápidamente.

[1] Es una señal de video que ha estado partida en dos o más componentes.

[2] Es una tecnología que se utiliza para permitir la visualización y audición de un archivo mientras se está descargando en el ordenador.

El problema aparecía a la hora de almacenar estos videos o transportarlos a través de la red de ordenadores pues ocupaban mucho espacio en disco. Esto propició la necesidad de la compresión de video digital. Los archivos comprimidos ahorran espacio, permiten una mayor portabilidad y son más fáciles de transferir a través de Internet. La compresión disminuye el número de parámetros requeridos para representar la señal, manteniendo una buena calidad perceptual[3].

Es importante decir que un video digital no es más que una secuencia de imágenes que ejecutadas a una determinada velocidad simulan movimiento. Por ello se hace necesario que las imágenes digitales también sean comprimidas, reduciendo la cantidad de datos necesarios para representarlas. La compresión de una imagen se basa en la eliminación de datos redundantes.

Hoy en día la compresión de imágenes es crucial pues ha jugado un papel importante en el crecimiento de la informática multimedia en temas como videoconferencias, imágenes médicas, televisión, control remoto de aplicaciones militares, entre otros. Aunque la compresión puede reducir la capacidad necesaria en el almacenamiento o transmisión de los datos de video, trae consigo pérdida en la información original y que la calidad del video baje, produciendo en ocasiones una serie de errores.

En el mundo del procesamiento visual se han establecido dos formas de evaluar la calidad de un video, la evaluación subjetiva y la evaluación objetiva. Hasta el momento la forma más segura y confiable para determinar esto es mediante la evaluación subjetiva, ya que los humanos son los receptores en la mayoría de las aplicaciones para estimar visualmente la calidad de la media. Un ejemplo de método de evaluación subjetiva de calidad es la puntuación media de opinión (MOS, Mean Opinion Score), la cual es obtenida a partir de un número de observadores. Este tipo de evaluación presenta desventajas como el empleo de mucho tiempo y recursos.

[3] Es la forma en la que el Sistema Visual Humano puede detectar a simple vista la transmisión del video sin errores.

Por estas razones se desarrolló la evaluación objetiva que busca alcanzar medidas de calidad precisas con un alto grado de correlación con la calidad subjetiva, proporcionando a través de cálculos matemáticos el índice de calidad de los videos y prediciendo la calidad perceptual de forma automática. Este tipo de evaluación se clasifica en tres grupos: métricas de detección de errores, distorsión estructural e híbridos. Este trabajo de diploma se enmarcará en la evaluación de calidad de los videos basado en la detección de errores. Pues la detección de errores evalúa la calidad de video, determinando la potencia de la señal de error y analizando en qué medida afecta a la percepción humana, según las características del SVH.

Actualmente a nivel mundial se le da mucha importancia al tema de la calidad de video. En Cuba también se trabaja para avanzar en este sentido y esto se percibe en la Universidad de las Ciencias Informáticas (UCI). Desde los inicios de la UCI fueron creados varios canales internos de televisión con fines educativos, culturales e informativos. Los procesos televisivos son controlados por la Dirección de Televisión Universitaria (DTU) y apoyado por el Centro Geoinformática y Señales Digitales (GEYSED) de la Facultad 6.

La Plataforma de Transmisión Abierta para Radio y Televisión (PTARTV) es un proyecto de innovación y desarrollo que se realiza en este Centro. Es una solución informática para la gestión y automatización de procesos radiales y televisivos. La misma integra y organiza los procesos que se realizan en dicha DTU. La plataforma cuenta con doce subsistemas, uno de los cuales es el Transferencia. Este brinda funcionalidades como es la conversión de las medias catalogadas al servidor, donde se almacenarán para luego ser publicadas y transmitidas. En muchas ocasiones estas medias presentan errores de conversión lo cual hace que las mismas no alcancen la calidad requerida para ser transmitidas. Esta situación provoca que se consuma más tiempo de lo previsto, o incluso, que en algunas circunstancias sean transmitidas con estos errores lo que causa molestias innecesarias a los televidentes. Por lo antes expresado se hace necesario desarrollar un sistema que permita evaluar la calidad de los videos convertidos en PTARTV basándose en métricas de detección de errores.

De la situación anterior se identificó el siguiente **problema de investigación:** ¿Cómo elevar el nivel de calidad de los videos trasmitidos por la plataforma PTARTV?

Para dar solución al problema planteado se define como **objetivo general:** Desarrollar un sistema que permita automatizar la evaluación de la calidad en los videos de la plataforma PTARTV, basado en métricas de detección de errores.

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 de que el desarrollo de un sistema para evaluar la calidad de los videos en la plataforma PTARTV, basado en métricas de detección de errores, permitirá mejorar la visualización de las transmisiones televisivas.

Para llevar a cabo la investigación, se definieron un conjunto de tareas que permitirán darle cumplimiento al objetivo planteado:

- Describir la evolución histórico-lógica de los procedimientos y técnicas para la evaluación de la calidad del video digital.
- Determinar las tendencias y tecnologías actuales más factibles para el desarrollo de la aplicación.
- Caracterizar procedimientos y técnicas de codificación y decodificación de video digital.
- Caracterizar el procesamiento de imágenes digitales basado métricas de detección de errores.
- Desarrollar los artefactos y documentación según la metodología de desarrollo seleccionada.
- Implementar las funcionalidades requeridas.
- Validar la propuesta de solución.

Para llevar a cabo un correcto desarrollo se utilizarán los siguientes métodos de investigación científica:

Métodos Teóricos:

Analítico–sintético: permitirá analizar y comprender la teoría y documentación relacionada con el chequeo de la calidad de las medias, facilitando así, extraer los elementos más importantes del objeto de estudio.

Análisis histórico–lógico: Permitirá entender el surgimiento y evolución del chequeo de la calidad de las medias, así como otras temáticas estrechamente relacionadas con la misma.

Modelación: permite una reproducción simplificada del fenómeno que se está estudiando. Se utiliza para crear abstracciones de los distintos procesos a desarrollar y así poder explicar la realidad.

Métodos Empíricos:

Observación: Facilita conocer el panorama real de una situación mediante la percepción directa del individuo. Con su utilización se determinaron los rasgos imprescindibles en el desarrollo de un sistema que permita evaluar la calidad de los videos convertidos en plataforma PTARTV.

Capítulo1: Fundamentación Teórica

1.1 Introducción

Para valorar correctamente la importancia del trabajo primeramente es necesario conocer acerca de los principales aspectos y conceptos relacionados con el dominio del problema. Por ello en este capítulo se describen una serie de conceptos abordados con frecuencia en la investigación. Además se especifica el objeto de estudio, haciendo una descripción general de la investigación.

Se tratan temas relevantes para la comprensión del contenido como son: imagen y video digital, el proceso de codificación, así como los tipos de evaluación de calidad existentes, entre otros. También se realiza un análisis de otras soluciones existentes con funcionalidades similares al sistema que desarrollará.

1.2 Conceptos asociados al dominio del problema

1.2.1 Plataforma de transmisión abierta para radio y televisión (PTARTV)

El producto PTARTV, provee informaciones mediante un sitio Web o un medio televisivo, haciendo uso de las nuevas tecnologías y logrando que se pueda acceder al contenido audiovisual en todo momento. Estas informaciones vienen asociadas a imágenes, sonidos y videos, procedentes de un repositorio de medias o de una transmisión en vivo. Con la implementación de esta aplicación se pretende dar solución a los problemas existentes en la Dirección de la Televisión Universitaria (DTU) o cualquier emisora de radio y televisión. Mediante funcionalidades que faciliten la programación, la gestión de las medias, la transmisión, la transferencia y el entorno web de la aplicación. (Ayala, 2010).

1.2.2 Video digital

Una vez digitalizado el video se representa como una secuencia de imágenes, donde se consideran un número adecuado por segundo que permitan crear en el espectador la ilusión de movimiento. El video es un buen ejemplo de verdadero formato multimedia, ya que típicamente permite combinar imágenes, audio y texto para crear secuencias animadas. (Manuel Agustí, 2010)

La digitalización del video proporciona grandes ventajas como son la posibilidad de almacenarlo, administrarlo y consultarlo. A pesar de esto presenta desventajas como el gran tamaño que ocupan en disco, por lo que trae dificultades en el almacenamiento y la transportación.

1.2.3 Imagen digital

Una imagen digital es una matriz bidimensional de niveles de grises, con elementos de información mínima, $g\,i/j$ que varían en función de la posición $\langle i, j \rangle$(fila, columna) que adoptan dentro de la matriz. Cada elemento dentro de la matriz se llama píxel. A cada píxel se le asigna un valor tonal, el cual está representado en un código binario. (García, 2002)

La clasificación de las imágenes puede realizarse basándose en diversos criterios, en este caso lo que interesa es solamente en cuanto a la forma en que una imagen se encuentra descrita en el ordenador. Se pueden distinguir dos grandes grupos de imágenes digitalizadas:

> **Imágenes vectoriales:** Las imágenes vectoriales están compuestas por entidades geométricas simples: segmentos y polígonos básicamente. Cada una de estas entidades está definida matemáticamente mediante precisas ecuaciones que describen perfectamente cada ilustración. (Ordoñez Ariza).
>
> **Imágenes bitmap:** Un mapa de bits es una matriz de bits que especifica el color de cada píxel en una matriz rectangular de píxeles. Cada uno de estos píxeles está relleno de un color uniforme, pero la sensación obtenida es el resultado de integrar visualmente, en la retina, las variaciones de color y luminosidad entre píxeles vecinos. (Microsoft, 2010)

1.2.4 Códec de video

Un códec[4] de video no es más que un conjunto de bibliotecas que permite comprimir y descomprimir video digital. Este utiliza un grupo de algoritmos, tratando siempre de reducir el tamaño del archivo. Usualmente los algoritmos de compresión que se emplean conllevan a una

[4] Códec: es la abreviatura de codificador-decodificador.

pérdida de calidad, por lo tanto siempre interesará utilizar los códec que logren más compresión y permitan pérdidas mínimas de calidad. (Alemañy, 2010)

1.2.5 Códec de audio

Un códec de audio es un tipo de códec específicamente diseñado para la compresión y descompresión de señales de sonido audible para el ser humano. Por ejemplo, música o conversaciones. Cumplen fundamentalmente la función de reducir la cantidad de datos digitales necesarios para reproducir una señal auditiva. Lo que comúnmente se denomina "compresión de datos", pero aplicado a un fin muy concreto. (Abreu, 2009)

1.2.6 Procesamiento digital de las imágenes

El procesamiento digital de video e imágenes consiste de un conjunto de algoritmos, métodos y técnicas, ejecutados en sistemas hardware/software, para el procesamiento de la información visual en medios donde se necesitan acciones como transformar imágenes de baja calidad en imágenes nítidas, producir efectos especiales, mezclar imágenes con gráficas y texto, comprimir la información, estandarizar el formato de video, corregir las no-linealidades de los dispositivos de entrada y salida, compatibilizar señales y equipos.(Campos, 2008)

1.2.7 Calidad

Se define calidad al conjunto de propiedades y de características de un producto o servicio, que le confieren aptitud para satisfacer unas necesidades explícitas o implícitas. (Sánchez, 2009-2010) Este concepto, llevado a imágenes o video, se entendería como la capacidad que una secuencia de video tiene de representar el objeto original, es decir, la exactitud o parecido entre ambos. (Baz Hormigos, 2009)

1.2.8 Métricas de calidad

Son un conjunto de algoritmos matemáticos que permiten evaluar los videos con calidad, que si se siguen correctamente pueden garantizar que el proyecto dará como resultado la satisfacción del cliente.

1.3 Objeto de estudio

En el presente trabajo se ha definido como objeto de estudio los procedimientos y técnicas para evaluar la calidad de videos basado en métricas de detección de errores.

1.3.1 Descripción general

Proceso de Codificación de video

La codificación de video permite comprimir la información facilitando su almacenamiento o transmisión con el menor tamaño posible mientras la calidad del video reconstruido satisface los objetivos de la aplicación. (Campos, 2008).(Ver figura 1)

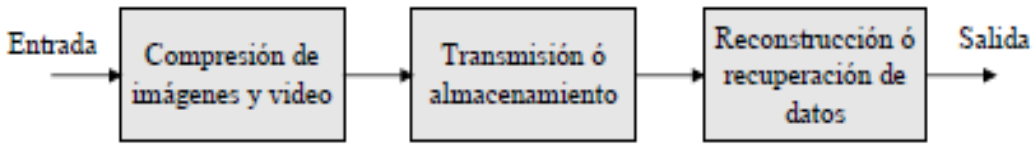


Figura 1: Proceso de codificación

(Campos,2008)

Los codificadores de video utilizan un conjunto de técnicas que normalmente son ejecutadas en cascada para proporcionar una solución de compresión óptima.

- Técnicas de compresión que no ocasionan pérdidas: debido al proceso de compresión se ejecuta removiendo la información redundante, la cual puede ser recreada por los datos que permanecen. Debido a esto la compresión puede ser revertida de forma exacta y precisa.

- Técnicas de compresión que producen pérdidas: Estas técnicas eliminan la información que no es importante para el usuario final, la cual no se puede recuperar. Debido a esto la compresión no se puede revertir totalmente.

Cada imagen de un video es dividido en macrobloques que cubren una imagen rectangular de 16x16 muestras de componente luma y 8x8 muestras de cada una de los dos componentes croma. Todas las muestras luma o croma de un macrobloque son predichas espacial o temporalmente y la predicción residual resultante es transmitida usando codificación de transformación. Por lo tanto cada componente de color de la predicción residual es subdividida en bloques y cada bloques transformado usando una transformación entera y los coeficientes son cuantizados y transmitidos usando métodos de codificación de entropía. (ABREU)

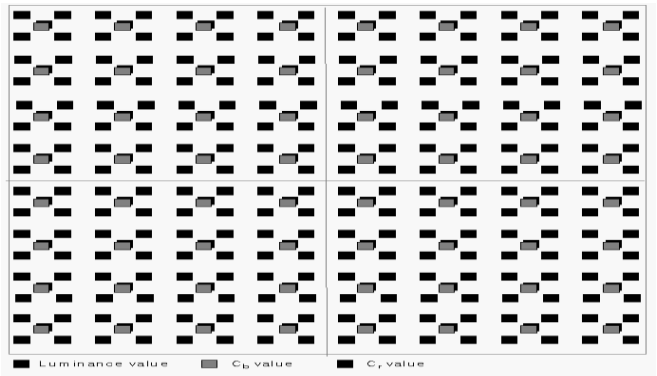


Figura 2: Macrobloque

(Carlos Esteban Baz Hormigos, septiembre,2009)

Un macrobloque no es más que el mínimo conjunto de bloques enteros de Y (luminancia) y Cr (crominancia) que ocupan la misma posición espacial en una imagen. (Ver figura 2). Los mismos constituyen la mínima unidad sobre la que se tratan los vectores de movimiento determinándose así el desplazamiento respecto a macrobloques con una información parecida. Estos macrobloques pueden estar divididos en bloques. Un bloque es la unidad mínima de tratamiento de imágenes constituido por un grupo de 8x8 muestras de Y, Cr o Cb de manera diferente. (Álvarez, 2010).

Degradaciones producidas en la compresión digital de videos:

Los algoritmos de compresión utilizados actualmente en la codificación digital de video introducen varios tipos de degradaciones, que se pueden clasificar según sus características principales. Esta clasificación es útil para poder comprender sus causas así como el impacto que tienen en la calidad percibida.

Ejemplo de algunos errores que se producen durante el proceso de conversión: (Joskowicz, 2008)

Efecto de bloques: Esta posiblemente sea la degradación que más se nota en un video digital. Tiene su origen en la codificación basada en bloques que segmenta la imagen en áreas pequeñas realizando una transformación de cada una de ellas de forma independiente. El efecto de bloques se presenta como discontinuidades en los bordes de bloques adyacentes al reconstruir la imagen. (Ver figura 3)

Figura 3: Ejemplo de efecto de bloques

Desenfoque o falta de definición: La falta de definición se manifiesta como una pérdida de detalle en la imagen. Si bien puede producirse por imágenes tomadas fuera de foco, también puede ser un efecto introducido por el proceso de digitalización. (Ver figura 5)

Figura 4: Ejemplo de desenfoque

Desplazamiento de color: El correspondiente efecto para la información de crominancia resulta en una contaminación de color en áreas con un nivel alto de contraste en crominancia. Se debe a la supresión de los coeficientes de alta frecuencia de los componentes de crominancia. Dado que la información de color es sub-muestreada, el efecto no se limita al bloque de 8x8 píxeles, sino que se propaga a todo el macro-bloque. (Ver figura 6)

Figura 5: Ejemplo de desplazamiento de color

Efecto Escalera: El efecto escalera se da al representar un borde diagonal como una concatenación de bloques horizontales y/o verticales.

Tipos de evaluación de la calidad de una imagen o video.

Como se expone anteriormente los sistemas de compresión muchas veces producen pérdidas en la calidad del video. Esto trae consigo la necesidad de evaluar en qué medida se produjo esta pérdida.

Hacer una evaluación significa estimar, apreciar, calcular el valor de algo. (RAE, 2010). Define en qué medida el elemento evaluado tiene unas características que se creen deseables, las cuales son especificadas a partir de la consideración de unos criterios.

Existen dos formas de evaluar la calidad de una imagen o una secuencia de video, mediante la evaluación subjetiva y mediante la evaluación objetiva.

La evaluación subjetiva de la calidad es la forma más confiable para determinar la calidad de una imagen o secuencia de video. Cuando se quiere evaluar de forma subjetiva cualquier tipo de evento, serán necesarias las opiniones que diferentes sujetos, los cuales deben ser

expertos en el tema en cuestión. Estos emiten una evaluación sobre dicho evento según criterios preestablecidos. Ejemplo de esto es la puntuación media de calidad (MOS) la cual es juzgada por un panel de observadores humanos, esta ha sido reconocida durante muchos años como el método más fiable de evaluación de la calidad. Sin embargo, este tipo de evaluación acarrea una serie de desventajas como son: el tiempo, los recursos necesarios, el coste y la imposibilidad de automatizar el proceso. (Carlos Esteban Baz Hormigos, septiembre,2009)

En algunas circunstancias es necesario que la valoración de la calidad sea independiente de las personas. Valorando las ventajas de los procedimientos automatizados, la comunidad científica ha realizado investigaciones en lo que se denomina evaluación objetiva de la calidad. Esta intenta eliminar la participación de los individuos, construyendo modelos teóricos y algoritmos de evaluación de calidad mayormente en base la distorsión percibida de la señal original. El propósito de esta es alcanzar medidas de calidad objetivas con un alto grado de correlación con la calidad subjetiva, de modo que se pueda predecir la calidad perceptual de forma automática. (Baz Hormigos, 2009)

Una medida objetiva de la calidad de video puede emplearse de tres maneras: (Baz Hormigos, 2009)

1. Para monitorizar la calidad de imagen en sistemas de control de calidad.

2. Para comparar sistemas de procesado de video. Si hubiera disponible múltiples sistemas de procesado para una tarea específica, entonces una medida de calidad puede ayudar a determinar cuál de ellos proporciona unos resultados con menor distorsión.

3. Puede ser integrada en el propio sistema de procesado de video para optimizar el algoritmo y los parámetros de configuración.

Las métricas de calidad objetiva se clasifican en tres grupos:

Distorsión estructural

La mayoría de los algoritmos para asignación de calidad consideran la imagen distorsionada como la imagen de referencia (perfecta) superpuesta con una imagen de ruido. La idea

de estos algoritmos es evaluar cuán perceptible es la señal de error para el sistema visual humano. (Casas Hernández, et al., 2005)

El problema es que el SVH[5] es sumamente complicado y el conocimiento existente sobre el mismo es limitado. Se introdujo una "nueva filosofía" en el diseño de métricas de calidad:

"La función principal del sistema visual humano es extraer información estructural del campo visual, y el sistema visual humano está altamente adaptado para este propósito. Por lo tanto, una medida de la distorsión estructural debería ser una buena aproximación a la distorsión percibida en una imagen". (Casas Hernández, et al., 2005)

Cuando se habla de estructura lo que se quiere decir es que existe una dependencia entre las muestras de una imagen, que aumenta a medida que estas muestras están más cercanas. A diferencia de los métodos basados en el SVH que considera diferencias píxel a píxel, en este método se considera el entorno que rodea al píxel. Otra ventaja del presente método es que se independiza (casi) del SVH y sus problemas actuales. (Casas Hernández, et al., 2005)

Híbridos

Los modelos híbridos de medida de calidad engloban modelos de calidad de imagen que tratan de dar una medida que se adecue a la percibida aunque no incorporan el modelo del SVH, explotan otras cualidades relativas a la percepción.

Detección de errores

Cuando surgió la detección de errores se concibió de forma de que no se incorporaran características propias del Sistema Visual Humano. Solo se basaban en funciones matemáticas simples, normalmente con un procesamiento en el dominio espacial y punto a punto de la imagen. El principio básico para la detección de errores es considerar la secuencia distorsionada como suma de una señal de referencia de calidad perfecta y una señal de error. Se crearon métricas como:

[5] SVH: Sistema Visual Humano

- MSE (“Mean Square Error”)
- PSNR (“Peak Signal to Noise Ratio”)
- NMSE (“Normalized Mean Square Error”)

Con el tiempo, los expertos en el tema se dieron cuenta de que necesitaban correlacionar las funciones matemáticas anteriormente utilizadas con las características del comportamiento del SVH. Pues las métricas con que contaban a pesar de que presentaban ventajas como su simplicidad y su velocidad, no tenían en cuenta las sensibilidades del sistema visual, trataban todas las distorsiones dándoles la misma importancia independientemente de su tipo, localización en la escena, etc. Por esto no determinaban errores que perfectamente pudieran detectar los observadores y por tanto no eran capaces de predecir de forma correcta la calidad que realmente era apreciada por el mismo. (Aguilar, 2006)

Actualmente existen métricas basadas en la detección de error que estiman la calidad perceptual midiendo los errores que pueden ser vistos por el SVH. Esto es posible por la simulación de la calidad perceptual relacionada con los componentes visuales. Sin embargo, el SVH es extremadamente complicado, un sistema altamente no lineal y no se conoce mucho acerca del tema. Por lo general las medidas de calidad que se encuentran fundamentadas en el SVH están muy vinculadas con la detección de errores.

Hasta hace poco tiempo los investigadores no hurgaban mucho en el tema de la evaluación de la calidad del video, se centraban más bien en el análisis de la evaluación de la calidad de las imágenes. Es importante decir que los algoritmos de medidas de calidad de video que se utilizan actualmente están fundamentas al igual que algoritmos de medidas de calidad de imágenes, en modelos del SVH con extensiones apropiadas para incorporar los aspectos temporales del SVH. Los algoritmos de evaluación de calidad de video son capaces de determinar la potencia de la señal de error y considerar la afectación que podría tener en la percepción humana. Una forma elemental de realizar una métrica de calidad de video es empleando un algoritmo de medida de imagen fija fotograma a fotograma. Sin embargo, una aproximación más sofisticada modelará los aspectos temporales del SVH. (Baz Hormigos, 2009)

Métricas basadas en la detección de errores

Medida de calidad de video (VQM, por sus siglas en inglés)

Es la Medida de Calidad de Video desarrollada por la Administración Nacional de Telecomunicación e Información (NTIA), dependiente de la Cámara de Comercio de los Estados Unidos. VQM es un método estandarizado de medir la calidad de video objetivamente. A pesar de esto es capaz de predecir muy cercanamente la calificación subjetiva de la calidad que se obtendría por un grupo de espectadores humanos. (Baz Hormigos, 2009)

VQM consiste en extraer características basadas en la percepción, calcular los parámetros de calidad de video y la combinación de parámetros para construir el modelo general. Esta métrica se basa en el uso de la DCT[6] para aprovechar las propiedades espaciotemporales del sistema de percepción humano. De igual forma, se basa en la propuesta de Watson, aunque le añade avances como: (Foncubierta Rodríguez)

- Aplicar dos matrices SCSF (Función de Sensibilidad a Contraste Espacial): una para imágenes fijas y otra para imágenes dinámicas, basadas a su vez en las matrices de cuantización MPEG. La diferencia entre ambas es una operación potencia proporcional a la tasa de fotogramas.

- Se estima el valor de la métrica a partir del máximo valor de distorsión en cada fotograma, ponderado con la media de distorsión. Según el autor, esto refleja el hecho de que una distorsión grande localizada enmascara la percepción de pequeñas distorsiones en el resto del fotograma.

Transformaciones a que somete la métrica: (Televisión, 2008)

- Se efectúa un muestreo y transformación de color que sirven para limitar el procesado a una región de interés y además para expresar la secuencia en términos de un espacio

[6] DCT: Transformada Discreta Coseno

de color perceptual. Este paso también se ocupa del desentrelazado y la corrección gamma de la señal de video.

- Las secuencias son sometidas a un bloqueo y a una transformada discreta del coseno y los resultados son entonces transformados a un contraste local.
- Se realiza el filtrado espacial y temporal y una operación de enmascaramiento de contraste.
- Finalmente las diferencias enmascaradas se muestran en dimensiones espacial, temporal y cromática para obtener la medida de calidad.

Esta medida es, actualmente, la que proporciona mejores resultados. Aunque como inconveniente tiene la carga computacional, que hace extremadamente compleja su implantación en sistemas de tiempo real.

Watson modificada

La medida de calidad de video digital de Watson (DVQ, Digital Video Quality) opera en el dominio de la DCT, la cual es muy eficiente en su implementación y la mayoría de los estándares de codificación. Esto hace la DVQ sea más atractiva desde un punto de vista del coste computacional. (Baz Hormigos, 2009)

Para la implementación del algoritmo DVQ es necesario realizar un conjunto de pasos, los cuales se enuncian a continuación:(Baz Hormigos, 2009)

- Calcular la DCT de las señales de referencia y la distorsionada.
- Computar el contraste local, aplica el filtrado para medir la función de sensibilidad al contraste (CSF) temporal y convierte los resultados a unidades Just Noticeable Difference (JND, por sus siglas en inglés) normalizándolos con los umbrales de visibilidad, tras lo cual la señal de error se computa.
- Aplicar el enmascaramiento y la combinación a las señales de error.

Es importante decir que DVQ implementa una transformación de color antes de aplicar la métrica a cada una de las dimensiones de crominancia.

Los resultados de la métrica se clasifican en:

[0 - 1.0] Buena
[1.1 – 2.0] Regular
[2.1 - 4] Mala
[4 - …] Muy Mala

Predicción de la puntuación de opinión media (MOSp)

Esta medida perceptual se basa en el uso del MSE[7] y su diseño está motivado por:

a) la obtención de una buena correlación con el MOS

b) mantener una carga computacional baja

Basándose en esta relación, se propuso la siguiente medida perceptual:

$$MOSp = 1 - ks \cdot MSE$$

Con ella se pretende estimar (predecir) la puntuación media de opinión (MOSp) de una secuencia comprimida, empleando el error cuadrático medio (MSE) entre los videos original y comprimido y la pendiente de la recta de ajuste (ks), que se calcula de forma automática a partir del contenido de la secuencia. (Baz Hormigos, 2009)

Después de una investigación se llegó a la conclusión que la métrica que sería factible es la DVQ, Digital Video Quality o Watson Modificada. Debido a que uno de los procesos de evaluación de la calidad de las medias en la plataforma PTARTV se realizará en tiempo real. En esta métrica se aceleran las operaciones de filtrado espacial el cual es uno de los procesos más complejos y costosos computacionalmente de este tipo de medidas. Para esto se usa la

[7] MSE: (Mean Squared Error):Error cuadrático medio

Transformada Discreta del Coseno en la descomposición en canales espaciales. Esto brinda una gran ventaja ya que en ocasiones la misma puede realizarse como parte del proceso de compresión.

1.3.2 Descripción actual del dominio del problema

La UCI es una institución que tiene como objetivo la formación integral de profesionales preparados, capaces de impulsar y promover la informatización de la sociedad cubana. La Universidad también pretende ser líder en la producción de software, apoyándose para esto en el trabajo que realizan los centros de desarrollo productivo. En la Facultad 6 de la Universidad se encuentra el centro de desarrollo GEYSED, en el cual se lleva a cabo el proyecto PTARTV. Esta plataforma responde a la automatización de los procesos que se llevan a cabo en la DTU y permite gestionar y automatizar los procesos radiales y televisivos en emisoras de radio y televisión. Dicha plataforma cuenta con doce subsistemas, uno de los cuales es el de Transferencia. Esta aplicación informática permite realizar la conversión de las medias para su publicación, la visualización para el chequeo de la calidad y el copiado hacia el servidor de la plataforma.

1.3.3 Situación problemática

Para realizar el proceso de conversión en la DTU el Centro de Gestión de Información Audiovisual (CGIA) destina una parte de su personal. Una vez realizado este proceso es necesario visualizar la media en busca de problemas o errores de conversión que atentan contra la calidad de la misma. Después de terminada la revisión si se presenta alguna degradación en la calidad del video debe reiniciarse el proceso de conversión. Esta situación provoca grandes pérdidas de tiempo y demoras que afectan a otros procesos en la DTU. En ocasiones, errores humanos cometidos en la visualización de los videos, traen como consecuencia transmisiones de poca calidad. Lo expresado anteriormente repercute en la opinión de los usuarios respecto al servicio televisivo que se le brinda.

1.4 Análisis de otras soluciones existentes

En la actualidad existen pocas soluciones informáticas que pudieran aportar al desarrollo de un sistema que evalúe la calidad de los videos en la Plataforma Abierta para radio y televisión,

permitiendo una mejor visualización de las transmisiones televisivas. Los sistemas que realizan estas funciones a nivel internacional están desarrollados con fines privativos, por lo que para utilizarlos sería necesario compralos y debido a que estos contienen una gran cantidad de funcionalidades son bastante costosos.

A nivel internacional existen herramientas que te permiten evaluar la calidad de los videos como por ejemplo el MSU (Video Quality Measurement Tool (VQMT)) que es un programa de medición de la calidad de video de forma objetiva. Esta aplicación te permite crear una comparación objetiva de los códec de video y realizar el proceso de análisis de video filtros.

Entre las principales características del programa se encuentran:

- Aplica funcionalidades flexibles pues es una herramienta que proporciona la funcionalidad para el cálculo: de valores de medición para cada marco, el valor promedio para cada frecuencia y los valores de medición para el componente de valores específicos.
- Soporte para 15 métricas de objetivo y más de 5 plugins. Entre ellas se encuentran PSNR, MSAD, SSIM, VQM, MSE, Delta, APSNR, 3SSIM y algunos plugins son: MSU Desenfoque métrico, MSU bloqueo métrico, MSU Brillo Sacudiendo métricas, MSU Brillo Independiente PSNR, MSU Drop métrica del sistema, MSU ruido Estimación Métrica, MSU cambio de escena Detector.
- Soporte para más de 30 formatos de video, incluyendo aquellos con 10, 14 y 16 bits de profundidad de color. Los formatos soportados incluyen AVI, YUV (YUV, YV12, UYVY, etc.), BMP (RGB24), AVS, MOV, VOB, WMV, MP4, MPG, MKV, FLV entre otros.
- Visualización Métrica: El programa puede generar una visualización de video por cada métrica especificada. Este video contiene la información visual de los valores de la métrica especificada de cada uno de los píxel del cuadro. La visualización del video permite tomar decisiones fácilmente sobre la naturaleza de la calidad del video y su visualización es generada por la combinación [métricas, componente de color, de referencia], el color y luminancia de cada píxel en video se define por el valor de métricas en ese píxel (muy útil para depurar y filtrar códec).

- Apoyo retorno de la inversión (a través de enmascaramiento): Permite calcular los parámetros en las Regiones de Interés (ROI) solamente. Retorno de la inversión debe ser especificado por siempre de video del additonal con la máscara en cada fotograma.
- Es capaz de comparar tres videos al mismo tiempo.

MSU es un software que fue desarrollado en dos versiones, una propietaria y una libre. La propietaria cuenta con una configuración muy buena, pero por problemas de licencia no se puede utilizar en el subsistema de transferencia de la plataforma PTARTV. El flujo de trabajo en este subsistema no es adaptable a la versión libre, ya que el software evalúa la calidad de las medias aplicando una series de métricas, las mismas para cuantificar el índice de calidad realiza comparaciones en los videos utilizando plugins, pero 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**.** Una de las funcionalidades que se debe implementar en la aplicación, es la evaluación automática, para esto es necesario acceder a la base de datos de la plataforma y buscar las direcciones de los videos original y comprimido, para luego cargarlos. Este proceso no es posible realizarlo con este sistema ya que su arquitectura solo permite cargar los videos manualmente. 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. Por estas razones no es posible utilizar esta versión en el subsistema de transferencia de la plataforma PTARTV.

Otra herramienta que permite evaluar la calidad del video es el software VQM (Video Quality Measurement PC) el cual proporciona un medio para evaluar objetivamente la calidad del video. El software realiza el proceso interactivo de fácil utilización de archivos de video. Se ejecuta bajo el sistema operativo Windows y proporciona dos métodos para realizar mediciones de calidad de video. En el primer método, el usuario debe tener almacenado en el disco el video original y el video procesado, luego el usuario simplemente crea una librería, alinea los archivos, analiza los que sean necesarios y calibra los videos. Determina espacial y temporalmente la región de interés, selecciona el modelo de calidad de video y las escalas subjetivas para seguidamente aplicar el cálculo matemático a través de la métrica VQM dándo como resultado los datos a medir para evaluar la calidad. En el segundo método, se supone que los usuarios tengan guardados en el disco los videos clip originales, donde más tarde se le

aplicará a los mismos un HRC (Circuito ficticio de referencia) como parte del proceso de evaluación. Ahora bien el proceso funciona cuando el usuario introduce los videos originales en una librería y seguido crea una prueba de secuencia de video (TVS) a los videos originales, después el usuario le aplica a la prueba de secuencia de video original un HRC y guarda el video procesado en el disco, después pasa por el flujo de calibración. Determina espacial y temporalmente la región de interés, selecciona el modelo de calidad de video y las escalas subjetivas para luego aplicar el cálculo matemático a través de la métrica VQM dándo como resultado los datos a medir para evaluar la calidad y así mide la calidad del video clip procesado con respecto al video original. (Margaret Pinson, 2000-2002)

En fin el flujo de este software se inicia con la creación de librerías, si las librerías contienen video original y procesado se va por el método uno y sino entonces se va por el método dos. Esta versión del software es privativa por lo que no se puede utilizar en el flujo de trabajo del subsistema de transferencia por problemas de licencia y configuración que la plataforma no es adaptable.

1.5 Conclusiones

En este capítulo se expusieron los principales conceptos asociados a la investigación que servirán de apoyo para el estudio, reflexión y entendimiento de la misma. Además servirá para la posible toma de decisiones y el correcto desempeño de la investigación en curso. Se realizó una descripción general de cómo elevar el nivel de calidad de los videos trasmitidos por la plataforma PTARTV. Describiendo detalladamente el dominio y situación problemática, permitiendo un mejor entendimiento para conocer de qué trata el problema de investigación .Se hizo además una caracterización de otras soluciones existentes que pudieran aportar al desarrollo de un sistema que permita automatizar la evaluación de la calidad en los videos basado en métricas de detección de errores. Arrojando como resultado, que los softwares para evaluar la calidad de los videos digitales encontrados no son adaptables ni configurables al flujo de trabajo de la plataforma. Por tanto esto impide que exista una herramienta que resuelva el problema de investigación planteado y siendo así se propone la creación de un sistema que permita automatizar la evaluación de la calidad en los videos basado en métricas de detección de errores. Por último, el estudio de las métricas basadas en detección de errores permitió arribar a la conclusión que el algoritmo más factible para su posterior implementación es DVQ.

Capítulo 2: Herramientas y tecnologías actuales

2.1 Introducción

En el presente capítulo se realizará un análisis de las herramientas y tecnologías que serán utilizadas en el desarrollo de la solución propuesta. Donde se seleccionarán las más factibles para el proceso de desarrollo del software. Se analizará las metodologías de desarrollo de software, la herramienta CASE[8] a utilizar, el entorno de desarrollo integrado (IDE), lenguaje de programación, librerías a incluir y otras herramientas necesarias.

2.2 Metodología de desarrollo de software

Las metodologías de software establecen cómo trabajar eficientemente evitando las catástrofes que conllevan al fracaso de un gran porcentaje de proyectos. Estas conforman un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de software. En ellas se van indicando paso a paso todas las actividades a realizar para lograr un producto informático robusto. A continuación se exponen las principales características de dos metodologías de desarrollo, Rational Unified Process (RUP por sus siglas en inglés) y Extreme Programming (XP por sus siglas en inglés).

2.2.1 El Proceso Unificado de Desarrollo de Software (RUP)

Siendo una de las metodologías de desarrollo de software más utilizada y robusta, RUP define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Utiliza el paradigma de orientación a objetos para su descripción. También constituye un marco de proceso configurable para satisfacer necesidades específicas, implementando las mejores prácticas de desarrollo de software. Es un proceso de software genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. Utiliza como lenguaje de modelado UML para preparar todos los esquemas de un sistema de software. Está dirigido por casos de uso ya que el proceso de desarrollo sigue un hilo, avanza a través de una serie de flujos de trabajos que parten de los casos de usos, se centra en la arquitectura, pues

[8] CASE(*Computer Aided Software Engineering*):Ingeniería de Software Asistida por Ordenador

incluye los aspectos estáticos y dinámicos más significativos del sistema. Además es iterativo e incremental, pues el trabajo se divide en pequeñas partes o miniproyectos, facilitando el control de los pasos en el flujo de trabajo y conllevando al avance y crecimiento del producto. (Jacobson, y otros, 2000)

El proceso unificado de desarrollo de un software conocido por sus siglas RUP divide el proceso en 4 fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto.

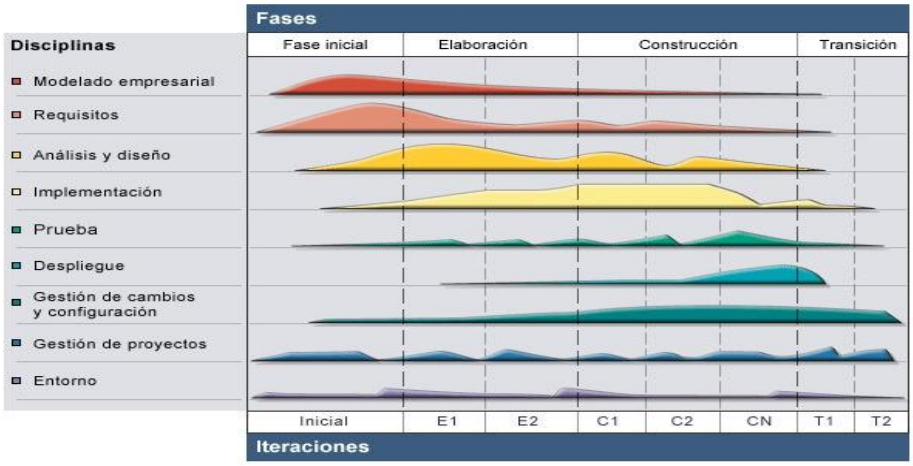


Figura 6: Fases y flujos de trabajo en RUP
(Corporación IBM, 2006)

En las iteraciones de cada fase se hacen diferentes esfuerzos en diferentes actividades:

- **Inicio:** se establece un caso de negocio para el sistema. Se deben identificar las entidades externas que interactuarán con el sistema y define estas iteraciones. (Somerville, 2005)
- **Elaboración:** permite una mejor comprensión del dominio del problema, establece un marco de trabajo arquitectónico para el sistema, desarrolla el plan de proyecto e identifica los riesgos clave del proyecto. (Somerville, 2005)
- **Construcción:** comprende el diseño del sistema, la programación y las pruebas. (Somerville, 2005)

- **Transición:** se ocupa de mover el sistema desde la comunidad de desarrollo a la comunidad del usuario y hacerlo trabajar en el entorno real. Al terminar esta fase se debe tener un sistema software documentado que funciona correctamente en su entorno operativo. (Somerville, 2005)

2.2.2 Extreme Programming (XP)

La programación extrema es posiblemente el método ágil más conocido y ampliamente utilizada. Está basada en una serie de valores y de prácticas de buenas maneras que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas. XP se centra en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo. Se preocupa por el aprendizaje de los desarrolladores y propicia un buen clima de trabajo. Se basa en la realimentación continua entre el cliente y el equipo de desarrollo, además de la comunicación fluida entre todos los participantes, la simplicidad en las soluciones implementadas y el valor para enfrentar los cambios. Es adecuada para proyectos con requisitos imprecisos, muy cambiantes y donde existe un alto riesgo técnico. Se recomienda emplearlo solo en proyectos a corto plazo. (Bautista)

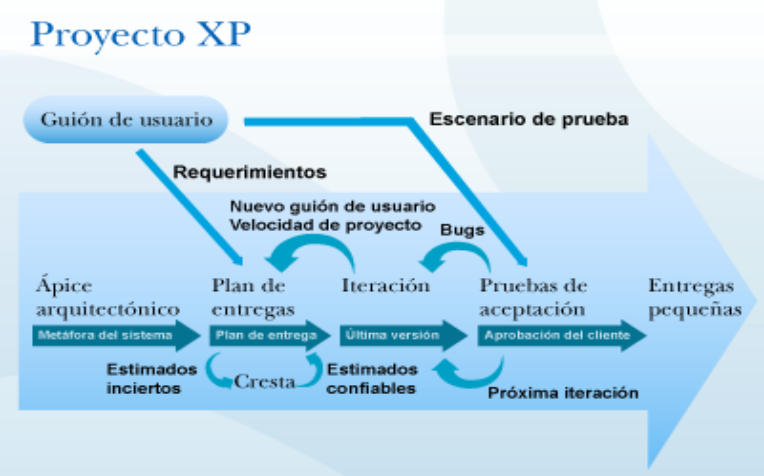


Figura 7: Proyecto con metodología XP

(Ayala, 2010)

Esta metodología está orientada a quien produce y usa el software. Algunos aspectos importantes de ella son: (Bautista)

- La comunicación, entre los usuarios y los desarrolladores.
- La simplicidad, al desarrollar y codificar los módulos del sistema.
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

2.2.3 Selección de metodología a utilizar

Luego de analizar estas dos metodologías, se concluyó que será RUP la metodología de desarrollo a utilizar. La misma se escoge por sus principales características y ventajas antes mencionadas. Además porque se adapta a las condiciones del proyecto mediante una previa configuración para su uso, realizando una configuración adecuada, que hace que el desarrollo del software sea rápido. Esta metodología permite controlar los posibles cambios que puedan ocurrir durante todo el proceso de desarrollo, define claramente los objetivos de cada fase, los entregables y sus actividades permitiendo un entorno en el que se agilicen las prácticas y técnicas necesarias para llevar a cabo la integración continua manteniendo diseños simples. Genera bastante documentación facilitando un mejor entendimiento tanto por parte del cliente como del grupo de desarrollo del software. El equipo de trabajo tiene experiencia en el uso de esta metodología, lo que hace que se desarrolle más fácil y rápidamente el producto. Propone varias iteraciones en el proceso de desarrollo de software posibilitando la corrección de errores y mejoras del software a medida que avanza el proyecto. Esto permite a los desarrolladores poder realizar un producto con gran calidad.

2.3 El Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado es un lenguaje estándar de modelado para software que permite visualizar, especificar, construir y documentar los artefactos que se crean durante el proceso de desarrollo del software. (Jacobson, y otros, 2000)

Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Es importante resaltar que no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. Es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos, es decir, describe cualquier sistema en términos de diagramas orientados a objetos. Es imprescindible destacar que un modelo UML describe lo que

supuestamente hará un sistema, pero no dice cómo implementar dicho sistema. (Booch, y otros)

Un modelo UML está compuesto por tres clases de bloques de construcción: (Booch, y otros)

- Elementos: son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)
- Relaciones: relacionan los elementos entre sí.
- Diagramas: colecciones de elementos con sus relaciones.

2.4 Fundamentación de la herramienta de modelado

Actualmente, muchas empresas se han extendido a la adquisición de herramientas CASE, con el fin de automatizar los aspectos claves de todo el proceso de desarrollo de un sistema. Esta herramienta comprende un amplio abanico de diferentes tipos de programas que se utilizan para ayudar a las actividades del proceso de software, como el análisis de requerimientos, el modelado de sistemas, la depuración y las pruebas. (Somerville, 2005)

Entre las ventajas que ofrece es que ayuda a la reutilización del software. También permite la portabilidad y estandarización de la documentación. Realiza gestión global en todas las fases de desarrollo de software con una misma herramienta. Además permite un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos.

2.4.1 Visual Paradigm

Visual Paradigm es una herramienta profesional que utiliza UML como lenguaje de modelado. Está dotado de productos o módulos que facilitan el trabajo durante la confección de un software en su ciclo de vida. Es un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación entre todos sus integrantes. (Mesa, y otros, 2010)

El software de modelado UML del Visual Paradigm ayuda a una rápida construcción de aplicaciones con buena calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML. Hace uso de un lenguaje común al equipo de desarrollo que facilita la comunicación y disponibilidad de diferentes versiones. Soporta que trabajen múltiples usuarios sobre el mismo

proyecto. Brinda facilidades para la exportación e importación de componentes. (Penas Rial, 2010)

Visual Paradigm es multiplataforma, diseñado para el trabajo tanto en Windows como en Linux. Está diseñado para una amplia gama de usuarios interesados en construir sistemas de software fiables con el uso del paradigma orientado a objetos, incluyendo actividades como ingeniería de software, análisis de sistemas y análisis de negocios. Es muy fácil de usar pues presenta un ambiente gráfico agradable e intuitivo para el usuario. Fácil de instalar y actualizar, está disponible en varias ediciones entre las que se encuentran Enterprise, Professional, Community, Standard, Modeler y Personal ofreciéndoles a los usuarios la posibilidad de escoger según sus necesidades.

2.4.2 Rational Rose Enterprise Edition

Rational Rose Enterprise Edition es una herramienta CASE que tiene la capacidad de crear, ver, modificar y manipular los componentes de un modelo. Crea los diagramas que se van generando durante el proceso de ingeniería en el desarrollo del software. Especifica, analiza y diseña el sistema antes de codificarlo. Posee facilidades para la generación de código a partir de diagramas previamente elaborados, donde utiliza como lenguaje de modelado UML. Se integra con otros IDES de desarrollo.

Facilita mecanismos para realizar la ingeniería inversa, es decir la realización de diagramas una vez conocido el código. Esta herramienta viene acompañada de un sistema de ayuda amigable. Se considera una de las herramientas más técnicas y de fácil uso. Es propietaria y no es soportada en el Sistema Operativo GNU/Linux razón por la cual es limitado su uso.

2.4.3 Selección de la herramienta de modelado a utilizar

Como herramienta de modelado visual se decide utilizar Visual Paradigm por UML Enterprise Edition v6.4 ya que soporta el ciclo de vida completo del desarrollo de software. La misma brinda ayuda a una rápida construcción de aplicaciones de calidad, a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Soporta que trabajen en un mismo proyecto múltiples usuarios. Tiene una ayuda que facilita el trabajo al usuario en caso de que no entienda como

usarla. Brinda facilidades para la exportación e importación de componentes. Es una herramienta multiplataforma que permite tanto el uso en Windows como en Linux. Esta herramienta está orientada al Lenguaje Unificado de Modelado influyendo en esto la posibilidad que ofrece la versión Community con su licencia gratuita. Además hay personal en el proyecto que cuentan con la experiencia necesaria, facilitando el proceso de entendimiento de la misma.

2.5 Framework de desarrollo

Un framework, en el desarrollo de un software, es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.Constituye una herramienta de extrema importancia para la implementación de aplicaciones en la actualidad. Proporcionan a los programadores y diseñadores una mejor organización y estructura de sus proyectos, lo cual acelera el desarrollo del producto final.

2.5.1 QT

El framework seleccionado es QT, pues proporciona varias herramientas y clases que permiten reducir el tiempo de desarrollo de una aplicación de escritorio. Es multiplataforma,ya sea su uso tanto en Windows como GNU\Linux. Permite desarrollar interfaces gráficas de usuario, integrándolo con el IDE de desarrollo Qt Creator. Distribuida bajo los términos de GNU License General Public License (y otras), Qt es software libre y de código abierto. Cuenta con un sistema de triple licencia: GPL v2/v3 para el desarrollo de software de código abierto y software libre y la licencia de pago Q Public License (QPL) para el desarrollo de aplicaciones comerciales. Una de las principales ventajas para los programadores es su extensa documentación disponible,acerca de las bibliotecas que brinda para ayudar en la implementación de cualquier aplicación. Además cuenta con una buena velocidad, debido a que está escrito en C++ y un buen diseño orientado a objetos que hacen muy placentero programar con Qt .Es fácil de utilizar y aprender,produce código legible, fácil de mantener y reutilizable. Demuestra un alto rendimiento en tiempo de ejecución y de pequeño tamaño. (Maldonado, 2011)

2.6 Lenguaje de programación

Es posible definir un lenguaje de programación como un conjunto de símbolos y reglas mediante las que se puede establecer la comunicación con el ordenador. De esta forma se puede controlar el comportamiento de una máquina, particularmente una computadora. De acuerdo a su nivel de abstracción, se habla de lenguaje de máquina, lenguaje de bajo nivel, lenguaje de medio nivel o lenguaje de alto nivel.

2.6.1 C++

C++ es un lenguaje de programación cuya intención al crearse fue el extender al exitoso lenguaje de programación C[9]. Se le añadieron funcionalidades de las que este último carecía, como la programación orientada a objetos (aunque no es un lenguaje enteramente orientado a objetos), manejo de excepciones y la sobrecarga de operadores. Es un lenguaje multipropósito que se adapta a múltiples situaciones y puede ser usado lo mismo para programación a bajo como a alto nivel. Aplica el uso de punteros. (Mora)

Ventajas: (Mora)

- Difusión: al ser uno de los lenguajes más utilizados en la actualidad posee un gran número de usuarios y existe gran cantidad de información, libros, cursos, páginas web que aceleran el proceso de entendimiento del mismo.
- Versatilidad: es un lenguaje de propósito general, por lo que se puede emplear para resolver cualquier tipo de problema.
- Portabilidad: el lenguaje está estandarizado y un mismo código fuente se puede compilar en diversas plataformas.
- Eficiencia: es uno de los lenguajes más rápidos en cuanto a ejecución.
- Herramientas: existe una gran cantidad de compiladores, depuradores y librerías.

[9] C:lenguaje de programación

Desventajas: (Mora)

Las principales desventajas del lenguaje C++ radican en que no todas las librerías disponibles para su uso son estándares de él y esto trae ciertos problemas de compatibilidad. El uso de ellas es complejo ya que el desarrollador debe encargarse de cargar y liberar de memoria estas librerías y correr los riesgos por el manejo de esta memoria. Es bastante complejo de aprender y no es recomendable para desarrollo de páginas web. También es necesario resaltar que por su uso de punteros puede ser algo más inseguro que lenguajes como Java o C#.

2.6.2 Java

Java es un lenguaje simple, orientado a objetos, distribuido, interpretado y compilado, robusto, seguro, portátil, de alto desempeño, de hilos múltiples y dinámicos. (Flanagan, 1998). Tiene una arquitectura neutra y funciona con uniformidad en gran cantidad de computadoras. Proporciona seguridad, al validar que los programas no violen las reglas semánticas del lenguaje. Facilita la administración del almacenamiento. Además simplifica la creación de estructuras dinámicas de datos al eliminar la necesidad de especificar explícitamente sus tamaños. (Savit, y otros, 1999)

Ventajas de Java: (Sanches, 2003)

- Es muy similar a los lenguajes C y C++, con las ventajas que tenían estos y con la virtud de facilitar su aprendizaje a todos los programadores de C
- Elimina los punteros, aumentando su seguridad
- Es totalmente orientado a objetos
- Está especialmente preparado para crear aplicaciones TCP/IP
- Implementa de forma nativa excepciones
- Es un lenguaje interpretado
- Control de tipo de datos más riguroso que en C
- Permite el uso de firmas digitales para asegurar la autoría
- Multihilo, es decir permite realizar varias tareas a la vez en el ordenador
- Es dinámico ,los objetos se cargan en memoria cuando son necesarios

Desventajas:

Las principales desventajas de Java son que depende para su funcionamiento de una máquina virtual, la cual para la interpretación del código, consume memoria RAM[10]. Esto es bastante complejo, dada las tareas que deberá asumir, puede disparar el consumo de recursos ralentizando además del sistema completo las tareas para las cuales fue destinado. Otra de sus desventajas es que no permite la sobrecarga de operadores, que limita mucho el uso que se les puede dar a los operadores dentro del programa teniendo que escribir mucho más en ocasiones donde una simple sobrecarga hubiera resuelto el problema. (Maldonado, 2009)

2.6.3 Selección del lenguaje de programación

Después de haber analizado las características de los lenguajes de programación mencionados previamente, se llega a la conclusión de utilizar el lenguaje C++. Este lenguaje cuenta con grandes potencialidades además de ser compatible en varias plataformas incluyendo GNU\Linux. Es un lenguaje de programación libre para la presentación de mayor velocidad de ejecución en comparación con otros lenguajes como Java. Su código fuente se puede compilar en diversas plataformas. Posee gran información, libros, cursos que aceleran el proceso de entendimiento del mismo. Su código es portable y puede ejecutarse en cualquier computadora y bajo cualquier sistema operativo. El entorno integrado de desarrollo QT Creator lo utiliza como lenguaje.

2.7 Fundamentación de los IDEs

Un entorno de desarrollo integrado o en inglés Integrated Development Environment (IDE) es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, puede utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación. Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los IDEs pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes.

[10] RAM (Random Access Memory): memoria de acceso aleatorio o memoria de acceso directo.

2.7.1 Qt Creator IDE

Qt Creator es un IDE para desarrollar interfaces gráficas de usuario y también para el desarrollo de programas sin interfaz gráfica como herramientas de la consola y servidores. Es desarrollado por la empresa Nokia, con licencias libres como la LGPL, por lo que se puede desarrollar con ella software abierto, libre, incluso comercial y privativo, sin comprar licencias. También poseen una licencia comercial para las empresas que quieran desarrollar software privativo. Cuenta aproximadamente con 500 clases, más de 9000 funciones y 500.000 líneas de código. Distribuida bajo los términos de GNU License General Public License (y otras), QT es software libre y de código abierto. (Nokia Corporation, 2011)

Qt Creator posee un avanzado editor de código C++, una GUI[11] integrada y diseñador de formularios. Además de resaltado y autocompletado de código. Cuenta con soporte para refactorización de código. Es multiplataforma ya que se ejecuta en Windows, Linux/X11 y Mac OS X sistemas operativos de escritorio. (Nokia Corporation, 2011)

2.7.2 Selección del IDE a utilizar

Se determinó que se va a utilizar el IDE QT Creator ya que por sus características puede ejecutarse sobre diferentes sistemas operativos como Windows y GNU\Linux. Además tiene licencias libres como la LGPL. Permite implementar programas con un código más legible y organizado y completa código agilizando el trabajo. También cuenta con una ayuda muy amplia, bien estructurada y ejemplificada, acelerando el proceso de entendimiento. Otro aspecto que influyó en la selección de este IDE fue el soporte que ofrece para el trabajo con la librería OpenCV y el lenguaje de programación C++ y todo tipo de códec.

2.8 Librerías incluidas

2.8.1 OpenCV

OpenCV (Open source Computer Vision library) es una biblioteca de código abierto desarrollada por la compañía Intel. Esta librería proporciona un alto nivel de funciones para el

[11] GUI: interfaz gráfica de usuario.

procesado de imágenes. Permite cargar imágenes, guardar imágenes, acceder a un píxel, gestionar imágenes, hacer transformaciones afines entre otras. Les facilita a los programadores crear aplicaciones poderosas en el dominio de la visión digital. (Calla y otros, 2007)

Implementa gran variedad de herramientas para la interpretación de la imagen. Incluye operaciones primitivas como binarización, filtrado, estadísticas de la imagen, entre otras. Implementa algoritmos para las técnicas de la calibración, detección de rasgos, análisis de algoritmos para las técnicas de la calibración, análisis del movimiento, reconstrucción 3D (Transformación de vistas), segmentación de objetos y reconocimiento. Los algoritmos están basados en estructuras de datos muy flexibles y usa la estructura Iplimage para crear y manejar imágenes. OpenCV se escribe en C y funciona bajo Linux, Windows y Mac OS X. (Calla y otros, 2007)

2.8.2 VTK

El Visualization ToolKit (VTK) es una biblioteca de clases, de código abierto, que permite programar en 3D completamente orientado a objetos. Con esta librería se pueden usar gráficos en 2D y 3D en varios tipo de lenguajes de programación (aunque esta hecho en C++), como Java, Python, C++. Está distribuido bajo el modelo Open Source, para computación gráfica, procesamiento de imágenes y visualización. Forma parte de las bibliotecas de alto nivel. (Buján, 2010). Dispone de una marco de visualización de información amplia, tiene un conjunto de widgets de interacción 3D, admite el procesamiento paralelo y se integra con varias bases de datos y GUI como Qt. Es multiplataforma ya que funciona en Linux, Windows, Mac y Unix. (Kiware, 2009)

2.8.3 Selección de la librería a utilizar

Después de haber analizado las características de las librerías expuestas se llegó a la conclusión que la librería a utilizar es la OpenCV ya que la misma facilita el trabajo con las imágenes proporcionando un alto nivel de funciones para el procesado de las mismas. Es compatible en varias plataformas por ejemplo Linux y Windows. Está escrita en el lenguaje C++ y proporciona varias herramientas y funciones que permiten reducir el tiempo de desarrollo de una aplicación de escritorio. Cuenta con una extensa librería de clases y herramientas para la

creación de elegantes aplicaciones y es soportado por el entorno integrado de desarrollo Qt Creator.

2.9 Sistema Gestor de Base de Datos.

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de herramientas que ayudan al usuario a gestionar información almacenada en una base de datos. (Gómez, y otros). Este permite definir los datos a los distintos niveles de abstracción. Además con él se pueden manipular los datos en la base de datos, pudiendo insertarlos, modificarlos, borrarlos y consultarlos. Manteniendo la integridad de la base de datos. También permite tener control de la privacidad y seguridad de los datos en la base de datos. (Cabello)

2.9.1 PostgreSQL

Es un sistema muy potente de gestión de base de datos objeto-relacional. Está basado en Postgres, desarrollado por la Universidad de Berkeley. Este se encuentra amparado por la licencia BSD (Berkeley Software Distribution), la cual permite el uso del código fuente en software no libre. Entre sus principales características están el aislamiento, la durabilidad y la atomicidad. Es importante decir que corre en GNU\Linux, Unix, Windows, entre otros sistemas operativos. Además existe mucha documentación, la cual está en diferentes idiomas, bien organizada, pública y libre.

Para el desarrollo de la solución se utilizará PostgresSQL, que es el sistema gestor de base de datos con el que se trabaja en la plataforma PTARTV. Este SGBD se escogió para el desarrollo pues es rápido, tiene buenas utilidades de administración, es confiable, robusto, fácil de usar tanto para volúmenes de datos grandes como pequeños y sin límites en los tamaños de registros. La conectividad, velocidad y seguridad, lo hacen altamente conveniente para acceder a bases de datos en internet ya que tiene mejor control de acceso a usuarios. Al mismo tiempo, los lenguajes que se proponen para el desarrollo de la plataforma logran una rica interacción con este gestor lo que resulta compatible y de mucho beneficio a la hora de diseñar de forma eficaz, aplicaciones dirigidas a bases de datos (Ayala, 2010)

2.10 Conclusiones

En este capítulo se realizó el estudio de las principales características de un conjunto de tecnologías, herramientas y tendencias asociadas a la posible propuesta, con el fin de solucionar el problema científico que concierne la presente investigación. Las principales metodologías para el desarrollo de software posibilitaron determinar a RUP como metodología a utilizar. 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 la librería 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

3.1 Introducción

Este capítulo tiene como objetivo lograr una mejor visión de la solución a través del sistema que se propone. Apoyándose para ello en los artefactos generados por la metodología de desarrollo de software. Se realiza la descripción de las principales definiciones asociadas al dominio del problema. Se construye el modelo de dominio para lograr un mayor acercamiento al problema a resolver. Se describen los principales conceptos de las clases del modelo de dominio, en un glosario de términos para su mejor entendimiento. Se hace un levantamiento de los requisitos tanto funcionales como no funcionales, se identifican los actores y casos de usos del sistema a implementar, así como sus respectivas descripciones textuales.

3.2 Principales conceptos y eventos del entorno

PTARTV es una plataforma que provee informaciones mediante un sitio Web o un medio televisivo, haciendo uso de las nuevas tecnologías y logrando que se pueda acceder al contenido audiovisual en todo momento. Estas informaciones vienen asociadas a imágenes, sonidos y videos, procedentes de un repositorio de medias o de una transmisión en vivo. (Ayala, 2010)

La plataforma cuenta con un servidor de **medias** que contiene películas, videos musicales, series, entre otras, que son sometidas a un proceso de conversión para luego ser publicadas y transmitidas a través de un sitio web o un medio televisivo.

El **proceso de conversión** se basa en aplicar técnicas de codificación a las medias que se quieran publicar. El principal objetivo de este proceso es lograr tener todas las medias almacenadas con un mismo formato y así lograr una mayor organización.

Las **medias procesadas** son el resultado de la ejecución del proceso de conversión las cuales serán publicadas para la transmisión a través de la web o medios televisivos.

En ocasiones estas medias procesadas pueden presentar **errores de conversión** que están dados por algún fallo en la codificación del video que se esté tratando. Esos errores pueden ser efecto de bloques, desplazamiento en los frame, desenfoque, distorsión del color y el sonido,

entre otros. A veces pueden ser detectados por el sistema visual humano y hacen que en ocasiones entorpezcan la transmisión y causen molestias en los televidentes o usuarios que estén interactuando con las medias.

Los errores mencionados anteriormente dan lugar a que las medias procesadas tengan que pasar por un **proceso de evaluación de calidad perceptual,** que consiste en detectar errores y evaluar la calidad de las medias aplicando una secuencia de pasos. Después de ser evaluadas se emitirá un reporte de esa evaluación, el cual mostrará los datos de la misma y así se sabrá si la media cuenta con la calidad necesaria para poder subirlas al servidor, de donde van a ser tomadas para la publicación y transmisión. Ese proceso se aplica utilizando **métricas de detección de errores** que utiliza algoritmos matemáticos para evaluar el índice de calidad de una imagen o una secuencia de video en específico.

La métrica de detección de errores a utilizar es **DVQ,** es la medida de calidad de video digital basada en algoritmos de detección de errores, que calcula para cada secuencia de video el índice de calidad, tomando una imagen original y otra procesada de referencia.

3.3 Modelo del dominio

Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un área de interés. Utilizando la notación UML se representa con un conjunto de diagramas de clases en los que no se define ninguna operación. Pueden mostrar objetos del dominio o clases conceptuales, asociaciones entre las clases conceptuales y sus atributos. Es una representación de las clases conceptuales del mundo real significativas en un dominio del problema, no de componentes de software, aunque algunos objetos del modelo pueden terminar siéndolo. (Larman)

El objetivo del modelo de dominio es realizar un análisis detallado de los objetos existentes en el entorno donde estará el sistema. Según RUP este tipo de modelo representa un subconjunto del Modelo de Objeto del Negocio, lo que no significa que siempre que haya un modelo de dominio tenga que existir obligatoriamente un modelo de negocio. (Jacobson, y otros, 2000)

Si se determina que no es necesario un modelo completo del negocio se realizará lo que se conoce como una modelación del dominio, permitiendo mostrar al usuario los principales

conceptos que se manejan en el entorno y de esta manera contribuir a la comprensión del contexto del sistema. Además de un glosario de términos para identificar todos los conceptos presentes.

3.3.1 Diagrama de clases del Modelo del dominio

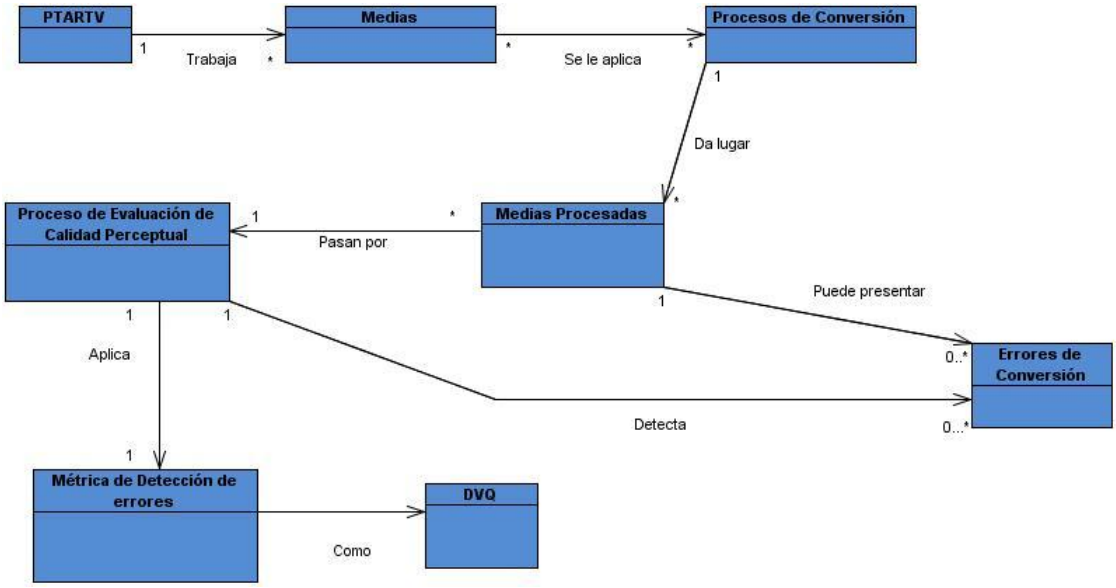


Figura 8: Modelo de Dominio

3.3.2 Glosario de Términos del Modelo del Dominio

El presente glosario de términos ayudará a una mejor compresión de los conceptos presentes en el entorno del problema.

PTARTV: Plataforma de Transmisión Abierta para Radio y Televisión es un Proyecto de Innovación y Desarrollo (I + D) que se lleva a cabo en el Centro Productivo GEYSED de la Facultad 6 en la UCI. Es una solución informática para la gestión y automatización de procesos radiales y televisivos, la cual permite integrar y organizar los procesos que se realizan en la Dirección de la Televisión Universitaria. Esta es capaz de difundir las informaciones mediante un sitio Web o señales televisivas, haciendo uso de las nuevas tecnologías, logrando que se pueda acceder al contenido audiovisual que pudiera emitirse por estos medios en todo

momento. Generalmente estas informaciones vienen asociadas a imágenes, sonidos y videos, normalmente procedentes de un repositorio de medias o de una transmisión en vivo. (Ayala, 2010)

Medias: se refiere a cualquier objeto que combina de forma simultánea varios tipos de contenidos, como pueden ser: video, audio, texto, imágenes, animación y otros. Aunque el término oficial es Multimedia, en numerosos medios de comunicación e informaciones oficiales se le llama también: “Media”. (Baby, y otros, 2009) Son todos los materiales audiovisuales tratados en la plataforma.

Proceso de conversión: maneja los ficheros que serán transmitidos o publicados en un formato estándar previamente seleccionado, así como codifica los materiales de audio y video a los formatos requeridos para la transmisión y publicación. (Dominguez Mora, 2009)

Medias procesadas: son el resultado de la ejecución del proceso de conversión, las cuales serán publicadas para la transmisión a través de la web o medios televisivos en la plataforma.

Proceso de evaluación de calidad perceptual: es el encargado de evaluar la calidad de las medias procesadas mediante una serie de pasos, brindando un reporte de la misma. Detecta disímiles errores que pueden ser vistos en ocasiones por el sistema visual humano y tiene como objetivo principal evitar que las medias convertidas no lleguen al servidor con errores, para luego ser transmitidas.

Errores de conversión: están dados por algún fallo en la codificación de la media que en ese momento este pasando por el proceso de conversión, por eso en ocasiones las medias procesadas presentan errores como por ejemplo: efecto de bloques, desplazamiento en los frame, desenfoque, distorsión del color y el sonido entre otros.

Métrica basada en detección de errores: permite a través de algoritmos matemáticos, evaluar el índice de calidad de una imagen o secuencia de video.

DVQ: algoritmo que calcula para cada secuencia de video el índice de calidad, tomando una imagen original y otra procesada de referencia.

3.4 Especificación de los requisitos de software

Según el glosario de términos de la IEEE se define como requisito de software:

"Condición o capacidad necesaria para que un usuario resuelva su problema o alcanza un objetivo que debe cumplirse o poseído por un sistema o componente del sistema para satisfacer un contrato, norma, especificación o formalmente impuesto en un documento". (Andrés González, 2010)

A continuación se especifican los requisitos funcionales y no funcionales, que el sistema debe ser capaz de cumplir. Para poder guiar el desarrollo del mismo cumpliendo con las exigencias de los clientes y alcanzar el éxito sin ningún percance.

3.4.1 Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir.

RF1 Validar Media

> **RF1.1** Escoger el tipo de evaluación de calidad.
>
> Este requisito permite escoger el tipo de evaluación de calidad, ya sea automática o manual.
>
> **RF1.2** Analizar región válida de procesado.
>
> Este requisito se encarga de verificar que se cumplan algunos parámetros imprescindibles para la evaluación de la calidad de las medias.

RF2 Evaluar la calidad de la media.

Este requisito permite evaluar la calidad de las medias convertidas mediante la comparación de la media original con la media procesada correspondiente.

RF3 Emitir reporte de calidad.

Este requisito mostrará el resultado de la evaluación realizada.

3.4.2 Requisitos no funcionales

Lo requisitos no funcionales se definen como las propiedades y restricciones del sistema.

Requisitos de usabilidad: La aplicación podrá ser usada por cualquier persona que posea conocimientos básicos en el manejo de los programas en la computadora y de un ambiente de escritorio en sentido general.

Requisitos mínimos de hardware:

- Core 2 Duo a 2.0 GHz
- HDD 30 GB
- Tarjeta de Red Gigabit Ethernet 1 Gbps
- 2 GB de Memoria RAM

Restricciones en el diseño y la implementación:

- Para el desarrollo de la aplicación se llevará utilizará el lenguaje C++, QT Creator IDE de desarrollo y la librería Open CV para el procesamiento gráfico de las imágenes.
- Se utilizará el lenguaje UML y la herramienta Visual Paradigm para el modelado del sistema.
- El sistema responderá a un modelo arquitectónico de tres capas.

Requisitos de apariencia o interfaz externa: La aplicación deberá ser sencilla, intuitiva, amigable, que le permita realizar acciones con fácil manejo y que posibilite también el buen desempeño en la ejecución de sus funciones.

Requisitos de portabilidad: El sistema estará realizado para funcionar en los sistemas operativos GNU/Linux o Windows, garantizando de esta manera un producto multiplataforma.

Requisitos de disponibilidad: El sistema debe estar disponible todas las horas del día.

Requisitos de confiabilidad: La información manejada por el sistema deberá estar protegida del acceso no autorizado y de divulgación a terceras personas.

Requisitos de rendimiento: El sistema debe mantener un rendimiento medio.

Requisitos de licencias: las herramientas en la que se desarrolla la plataforma son libres.

Requisitos legales, de derecho de autor y otros: sus derechos de autor y otros están determinados por la empresa comercializadora del producto, ALBET S.A y la entidad desarrolladora, la UCI.

3.5 Descripción del sistema propuesto

Una vez realizado el modelo de dominio e identificados los requerimientos del software es posible modelar y proponer el modelo del sistema de la aplicación, el cual estará enmarcado en garantizar la evaluación de la calidad de las medias procesadas en la plataforma.

3.5.1 Descripción de los actores

Actores	Descripción de los actores
Conversor	Cualquier persona que autorizada haga uso del subsistema transferencia de la PTARTV o sencillamente necesite evaluar la calidad de las medias convertidas.

Tabla 1. Descripción de los actores del sistema

3.5.2 Diagrama de casos de uso del sistema

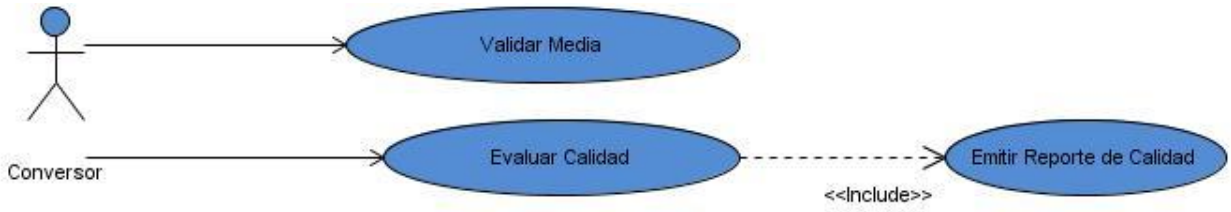


Figura 9: Diagrama de casos de uso del sistema

3.5.3 Descripción de los casos de uso del sistema

Un caso de uso especifica el comportamiento de un sistema o de una parte del mismo y es una descripción de un conjunto de secuencias de acciones, incluyendo variantes, que ejecuta un sistema para producir un resultado observable de valor para un actor. Proporcionan un medio para que los desarrolladores, usuarios finales del sistema y los expertos del dominio lleguen a una comprensión común del sistema. Además, ayudan a validar la arquitectura y a verificar el sistema mientras evoluciona a lo largo del desarrollo. Describe un conjunto de secuencias, donde cada secuencia representa en la interacción de los elementos externos, a sus actores con el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Además se utilizan para ilustrar los requerimientos del sistema, al mostrar cómo reacciona una respuesta a eventos que se producen en el mismo. Se consideran 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. (Booch, y otros)

A continuación se describirá los casos de usos del sistema (DCU).

3.5.3.1 Evaluar calidad

Caso de Uso	Evaluar calidad
Actores	Conversor
Propósito	Este caso de uso se lleva a cabo con el objetivo de que el conversor pueda calcular el índice de calidad del video procesado.
Resumen	El caso de uso se inicia cuando la validación de los videos finalizó y finaliza con la confirmación de que el video se evaluó.
Precondiciones	Debe haberse analizado la región válida de procesado.
Referencias	RF2
Prioridad	Crítico

<table>
<tr><th colspan="2">Flujo Normal de Eventos</th></tr>
<tr><th>Acción del Actor</th><th>Respuesta del Sistema</th></tr>
<tr><td>1. El caso de uso se inicia cuando el sistema termina de validar las secuencias y luego procede a la evaluación de la misma.</td><td>1.1 El sistema ejecuta alguna de las siguientes opciones:
• Si en el caso de uso “Validar media” en el paso 1.1 del flujo normal de los eventos, el usuario escoge la opción “Manual” va a la sección “Proceso Manual”.
• Si en el caso de uso “Validar media” en el paso 1.1 del flujo normal de los eventos, el usuario escoge la opción “Automático” va a la sección “Proceso Automático”.</td></tr>
<tr><th colspan="2">Acción del Actor</th></tr>
<tr><th colspan="2">Sección “Proceso Manual”</th></tr>
<tr><th>Acción del Actor</th><th>Respuesta del Sistema</th></tr>
<tr><td>2. El conversor accede a la interfaz para realizar la evaluación de las medias manualmente.</td><td>2.1 El sistema muestra una interfaz con el botón:
• Evaluar.
• Cancelar
• Barra de Progreso</td></tr>
<tr><td>3. El conversor presiona el botón Evaluar.</td><td>3.1 El sistema muestra barra de progreso que indica que se está evaluando la calidad de la media.
3.2 El sistema finaliza la evaluación de la calidad satisfactoriamente.</td></tr>
</table>

Prototipo de Interfaz	
	
Flujos Alternos	
	Respuesta del Sistema
4. El actor procede a anular la evaluación presionando el botón Cancelar.	4.1 El sistema cancela la operación.
Poscondiciones	Se realiza la evaluación de la calidad de la media.
Flujo Normal de Eventos	
Acción del Actor	
Sección "Proceso Automático"	
Acción del Actor	**Respuesta del Sistema**
1. El conversor accede a la interfaz para realizar la evaluación de las medias automáticamente.	1.1 El sistema muestra una interfaz con los siguientes campos, permitiendo entrar los datos necesarios para poder conectarse a la base de datos y cargar las direcciones de las medias procesadas y original.

	• IP • Puerto • Usuario • Contraseña • Base de Datos • Botón Conectar • Botón Cancelar
2. El actor presiona el botón “Conectar”.	2.1 El sistema se conecta a la base de datos y carga las direcciones correspondientes.
3. El conversor presiona el botón “Evaluar”.	3.1 El sistema inhabilita los botones que permiten seleccionar la dirección de las medias de forma manual y carga la dirección de la secuencia original y procesada automáticamente. 3.2 El sistema procede a validar internamente las secuencias cargadas y seguidamente a evaluar la calidad de las mismas.
	3.3 El sistema muestra una barra de progreso que indica que se está realizando el proceso de evaluación, finalizando así este caso de uso.
Prototipo de Interfaz	

Flujos Alternos	
Acción del actor	**Respuesta del Sistema**
3.4 El actor procede a anular la evaluación de la calidad presionando el botón Cancelar.	3.5 El sistema cancela la operación.
Poscondiciones	Se realiza la evaluación de la calidad de la media.

Tabla 2. DCU Evaluar Calidad

3.6 Conclusiones

En este capítulo se realizó el modelo de dominio el cual se utilizó para lograr la compresión de los conceptos utilizados por los usuarios y con los que deberá trabajar la aplicación. Se analizaron y valoraron todos los requisitos de software planteados por los usuarios finales del sistema. Como resultado principal de este capítulo quedó plasmada la propuesta de solución para el problema actual del sistema, detallada a través de los actores, casos de usos, diagrama de casos de uso, así como las especificaciones de los casos de uso,creando una base para sustentar las siguientes fases de desarrollo.

Capítulo 4: Construcción de la solución propuesta

4.1 Introducción

En el presente capítulo se abordarán aspectos fundamentales asociados a la construcción de la solución propuesta. Se definirá la arquitectura del sistema, haciendo énfasis en los patrones de arquitectura utilizados para su realización. Se describirá las principales características de los patrones del diseño identificados. Además se desarrollará el diseño del sistema, donde se elaborará el diagrama de clases del sistema, el modelo del despliegue, diagramas de componentes, así como una fundamentación teórica de cada modelo presentado. Se realizará las pruebas a la solución presentada, para su validación.

4.2 Definición de la arquitectura de software

La arquitectura del software desempeña un papel importante durante el desarrollo de un sistema. La misma permite organizar sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución. Expresan un esquema organizativo estructural para sistemas de software y definen las reglas generales de organización, las restricciones en la forma y la estructura de un grupo numeroso de aplicaciones. La selección de un patrón arquitectónico es una decisión básica del diseño en el desarrollo de un sistema. Para el desarrollo de las funcionalidades a implementar se utiliza el patrón arquitectónico Capas, que define un modelo general de N-capas para la arquitectura lógica. Las ideas esenciales del patrón Capas se centran en: (Larman)

- Organizar la estructura lógica de gran escala de un sistema en capas separadas de responsabilidades distintas y relacionadas, con una separación clara y cohesiva de intereses. Las capas "más bajas" son servicios generales de bajo nivel y las capas más altas son más específicas de la aplicación.

- La colaboración y el acoplamiento es desde las capas más altas hacia las más bajas; se evita el acoplamiento de las capas más bajas a las más altas.

Según Pressman, en su libro “Ingeniería de Software. Un enfoque práctico”: la arquitectura de tres capas permite aislar a la tecnología que implementa la base de datos, de forma que sea

fácil cambiar esta tecnología. Utiliza mucho código lejos del cliente y los cambios de mantenimiento ocurren de forma centralizada. La idea de las tres capas encaja con las prácticas orientadas a objetos de hoy en día: todo el procesamiento tiene lugar por medio de los mensajes que se envían a los objetos y no mediante trozos de código asociados a cada objeto."

La capa de presentación: es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser amigable para el usuario generalmente se presentan como formularios. (2011)

La capa de negocio: es donde se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él. (2011)

La capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. (2011)

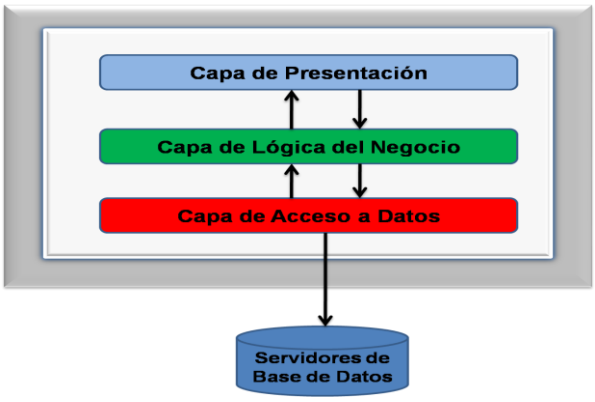


Figura 10: Arquitectura en Capas

Se selecciona la arquitectura anteriormente mencionada pues la misma facilita la modularidad del sistema, la localización de errores y mejora considerablemente el soporte del mismo. Cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior y las interacciones entre estas ocurren generalmente por invocación de métodos. Además permite el desarrollo paralelo del sistema por cada capa. Facilita el mantenimiento y soporte del proyecto. Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad). Con el uso del modelo de tres capas se logra en cierta medida obtener una mejor organización durante la realización del software.En dicha arquitectura a cada nivel se le confía una misión concreta, lo que permite el diseño de una arquitectura escalable (que pueden ser ampliada con facilidad en caso de ser necesario).

El uso de esta arquitectura se evidencia en el sistema propuesto, separando las clases necesarias para el funcionamiento del mismo y estructurandolas de forma que, cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. Para ello en la capa presentación se encontrarán las clases interfaces, por ejemplo la clase interfaz Evaluar_Calidad, Mostrar_Reporte y Config. Estas serán las clases con las cuales los usuarios interactuarán en la aplicación. En la capa lógica del negocio, estarán las clases que permitirán realizar el proceso interno del sistema. Es decir es donde se tomará las solicitudes que envía el usuario y esta capa será la encargada de darle un resultado. Esto lo procesará a través de las relaciones y funcionamiento entre las clases:Cdvq_Control,CVideo,Cimagen,CBloque,CGestionar_Reporte,CHilo_Manual,CHilo_Automático,CLector_ffmpeg, que permitirán realizar el proceso de evaluación de la calidad de los videos. Además puede comunicarse esta capa con la de acceso a datos para buscar algún dato necesario en la base de datos. La capa de acceso a datos es la encargada de acceder a los datos,aquí estarán las clases CDatos y CConexión.

4.3 Principios del diseño

Según Pressman los principios básicos del diseño hacen posible que el ingeniero del software navegue por el proceso de diseño. Estos proporcionan un marco de trabajo necesario para lograr que se realice correctamente. Favorecen la gestión de la complejidad de los sistemas de software y la obtención de los factores de calidad que estos sistemas han de presentar. Sugiere

un conjunto de principios para el diseño del software los cuales serán mencionados a continuación:

- En el proceso de diseño no deberá utilizarse orejeras
- El diseño no deberá inventar nada que ya esté inventado
- El diseño deberá presentar uniformidad e integración
- El diseño deberá estructurarse para admitir cambios
- El diseño deberá evaluarse en función de la calidad mientras se va creando, no después de terminado
- El diseño deberá revisarse para minimizar los errores conceptuales

4.3.1 Estándares de la interfaz de la aplicación

La aplicación debe presentar una interfaz sencilla, amigable y eficiente ya que este 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.

4.4 Patrones del Diseño

Los patrones de diseño, constituyen soluciones estándar que brindan respuesta a un problema común en materia de diseño de sistemas e implementación. Estos permiten establecer una gran comunicación entre los diseñadores, facilitando además el aprendizaje del programador inexperto y logrando crear parejas problema-solución. Cada patrón sugiere además numerosas ventajas, dentro de ellas la reutilización de código y el permitir la realización de un diseño apto para el cambio. (Barrios Cardoso, et al., 2010)

Según Craig Larman se define patrón de manera más simple, “a un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos”.

A continuación se presentarán los diferentes patrones identificados en el proceso de modelado de las clases del diseño, patrones GRASP (General Responsibility Assignment Software Patterns) y GoF (Gang of Four).

4.4.1 Patrones GRASP

Según Craig Larman *los patrones describen principios fundamentales del diseño de objetos y la asignación de responsabilidades.*

GRASP es un acrónimo de General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). El nombre se eligió para sugerir la importancia de aprender estos principios para diseñar con éxito el software orientado a objetos. (Larman).Para asignar la responsabilidad a un objeto determinado se sugiere el uso de patrones de diseño GRASP, entre los que destacan:

Experto: Se utilizará este patrón para la asignación de responsabilidades indicando que la clase que cuenta con la información necesaria para cumplirla, es la encargada de manejar la información. Expresa simplemente la intuición de que cada clase contiene los métodos relacionados con la información que posee. Este patrón se aplicó a las clases entidades Gestionar_Reporte, CVideo, CImagen y CBloque. Estas clases son las únicas que conocen de los detalles que ocurren en cada una, por ejemplo la clase CReporte va a ser la encargada de gestionar los datos necesarios para emitir un reporte. La clase CVideo, será la encargada de aplicar el DVQ total al video deseado, lo mismo ocurre con la clase CImagen que es la encargada de realizar el procesado de bloques y en la clase CBloque es la que realiza los pasos necesarios que debe aplicar la métrica como por ejemplo DCT,contraste local,conversión a JND.

Creador: Se utilizará este patrón dado las propiedades de todo sistema orientado a objetos. Se aplicará en los casos donde una clase tiene la responsabilidad de crear una nueva instancia de la otra. Se puede ver el uso de este patrón en el diseño de la clase controladora CC_Cdvq_Control que la misma crea una instancia video y una instancia reporte. También en la clase CVideo, se crea una instancia imagen y en la clase CImagen se crea una instancia bloque.

Controlador: Define quién deberá encargarse de atender un evento del sistema. Es una clase que, para el diseñador representa de alguna manera al sistema global. Este patrón se evidencia a través de la clase controladora CC_Cdvq_Control la cual se encarga de los eventos más significativos del sistema.

Alta cohesión: Se ocupa de que las clases del diseño realicen las funcionalidades necesarias para cumplir con las tareas que tienen definidas. Plantea la contribución entre clases para realizar tareas de elevada complejidad. Este patrón se evidencia en todas las clases del diseño del sistema DVQ.

4.4.2 Patrones GoF

Los patrones de diseño GoF son conocidos como la pandilla de los cuatro. A continuación se mencionan los patrones que fueron aplicados directamente en el diseño del sistema.

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.

Composición: este patrón es usado cuando se pretende que los clientes no reparen en las diferencias entre objetos simples y compuestos. 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.

4.5 Modelo del Diseño

El modelo del diseño es un modelo de objetos que describe la realización física de los casos de usos centrándose en cómo los requisitos funcionales y requisitos no funcionales , junto con otras restricciones relacionadas con el entorno de la implementación ,tienen impacto en el sistema a considerar. Sirve de abstracción de la implementación del sistema y es de ese modo utilizada como una entrada fundamental de las actividades de implementación. (Jacobson, et al., 2004)

4.5.1 Diagramas de Clases del Diseño

El Diagrama de Clases del Diseño (DCD) muestra el sistema en términos de clases y métodos. Muestra cómo quedaría implementada toda la aplicación en términos lógicos.

Modelo General de las Clases del Diseño

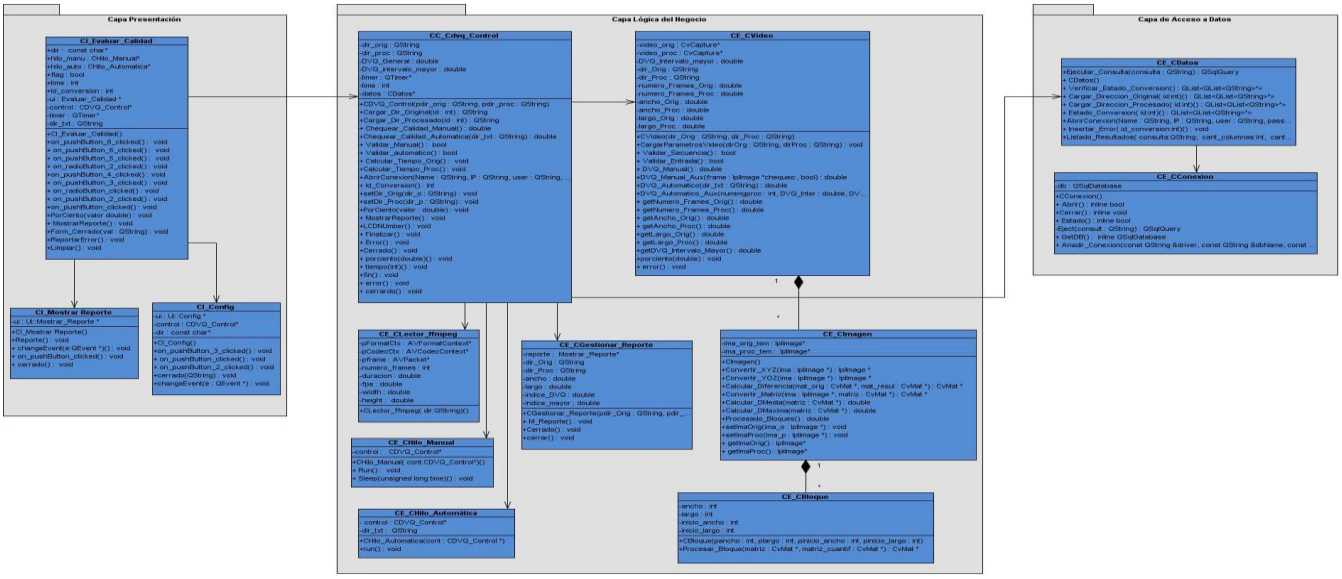


Figura 11: Diagrama General de Clases del Diseño

4.6 Modelo de Despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo .Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño. (Jacobson, et al., 2004)

4.6.1 Diagrama de Despliegue

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación que muestra las relaciones físicas entre los componentes hardware y software en el sistema propuesto .A continuación se muestra el diagrama de despliegue.

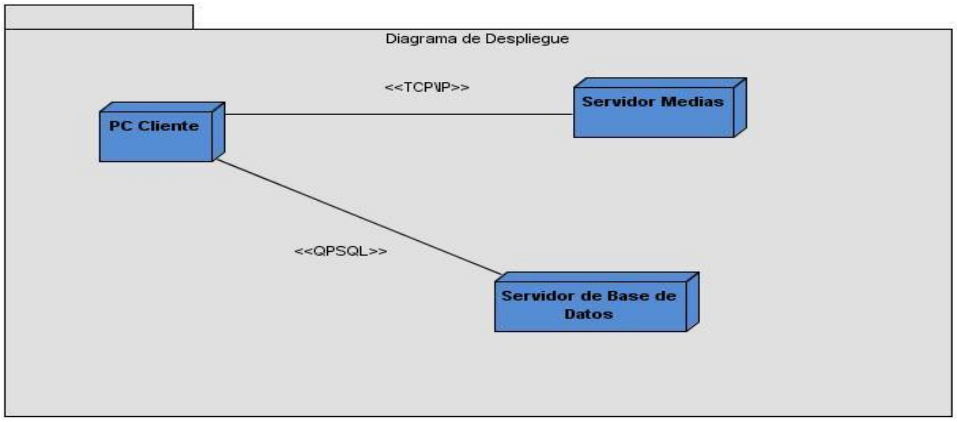


Figura 12: Diagrama de Despliegue

A continuación se describen todos los elementos que componen este diagrama:

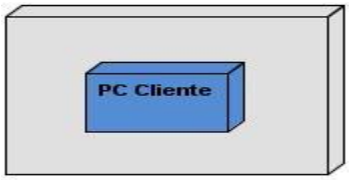


Figura 13: Nodo PC Cliente

Nodo PC_Cliente: Su función es acceder al sistema para evaluar la calidad de las medias e interactuar con el mismo según sus necesidades. Su conexión es a través del Protocolo de Comunicación TCP\IP al servidor donde estén las medias y se conectará a la base de datos a través de la librería QPSQL.

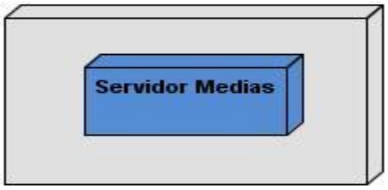


Figura 14: Nodo Servidor de Medias

Nodo Servidor de Medias: En este nodo se almacenarán las medias que cuentan con la calidad requerida para ser transmitidas posteriormente en la plataforma.

Figura 15: Nodo Servidor de Base de Datos

Nodo Servidor de Base de Datos: En este nodo estará corriendo el servidor de Base de Datos PostgresSQL, mediante el cual se gestionará la información necesaria para evaluar la calidad de las medias en la plataforma.

QPSQL: librería que es utilizada para conectarse a la base de datos, con la que va a trabajar el sistema.

TCP\IP: es un conjunto de protocolos de red en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras.

4.7 Modelo de Implementación

El modelo de implementación describe cómo los elementos del modelo del diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables. Describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles, en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y como dependen los componentes unos de otros. (Jacobson, et al., 2004)

4.7.1 Diagrama de Componentes

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Un componente es el empaquetamiento físico de los elementos de un modelo como lo son las clases del diseño. (Jacobson, et al., 2004). Para una mejor comprensión se mostrará la vista general del diagrama de componentes y las relaciones que existen entre sus capas y componentes establecidos.

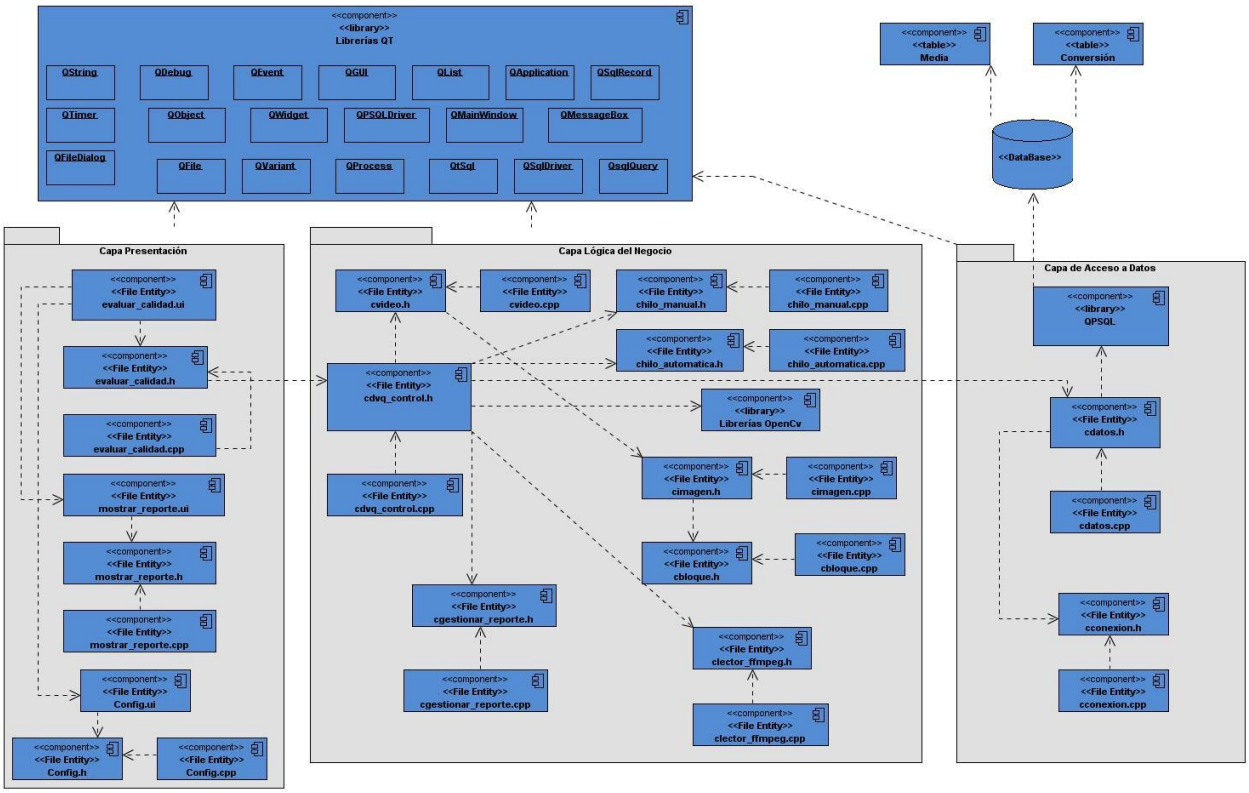


Figura 16: Diagrama de Componentes

4.8 Prueba del sistema propuesto

En el desarrollo de cualquier software, las pruebas son de vital importancia ya que no es más que procesos que permiten verificar y revelar la calidad del mismo. Son utilizadas para identificar posibles fallos que pueden estar relacionados con la implementación, calidad, o usabilidad del sistema. Cada prueba tiene su estrategia y propósito. La ausencia de defectos no puede ser asegurada a través de las pruebas, sino que solo se puede demostrar que existen errores en el software. Con el objetivo de validar las funcionalidades del sistema se realizarán pruebas funcionales y unitarias para lograr buena calidad en el producto.

4.8.1 Pruebas Funcionales

Se denominan pruebas funcionales, a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados. Se basan en la ejecución, revisión y retroalimentación de las funcionalidades

previamente diseñadas para el software.La misma toma el punto de vista del usuario y sus funciones son probadas ingresando datos de entradas y examinando datos de salida. Son denominadas también pruebas de caja negra.

Las pruebas de caja negra son las que se llevan a cabo sobre la interfaz del software, por lo que 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 una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software. Se centran principalmente en los requisitos funcionales del software.

Muchos autores consideran que estas pruebas permiten encontrar: (Pressman, 2002)

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

4.8.1.1 Casos de Pruebas

Los casos de pruebas permiten detallar la forma en la que se va a validar las funcionalidades y algoritmos utilizados para el desarrollo del sistema. Es un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados, desarrollados para cumplir un objetivo en particular o una función. Es la entidad más simple que siempre es ejecutada como una unidad, desde el comienzo hasta el final.

Los casos de pruebas deben verificar:

- Si el producto satisface los requerimientos del usuario, tal y como se describe en las especificación de los requerimientos.

- Si el producto se comporta como se desea, tal y como se describe en las especificaciones funcionales del diseño.

A continuación se muestra uno de los casos de pruebas realizados en el sistema. Decir que a medida que se fueron realizando las pruebas a las distintas funcionalidades del sistema, se pudo apreciar como resultado algunos errores, que posteriormente fueron corregidos satisfactoriamente. Los demás casos de pruebas remitirse a los anexos.

Caso de prueba	**Evaluar calidad**
Condiciones	Debe haberse validado las secuencias a comparar en el proceso de evaluación de la calidad de las mismas.
Datos de entrada	Secuencias (original, procesada) para evaluar la calidad.
Resultados esperados	1. Debe quedar seleccionado el tipo de evaluación.
	2. Debe haber seleccionado las secuencias a evaluar calidad.
	3. Debe brindar el índice de calidad de la media procesada.
Resultados obtenidos	4. La prueba es satisfactoria. El tipo de evaluación y las secuencias a evaluar son seleccionadas por el usuario.
	5. La prueba es satisfactoria. Brindando el índice de calidad de la media procesada.
Observaciones	Al realizar la prueba en el sistema se obtuvo el resultado correcto.

Tabla 3 : Caso de Prueba Evaluar Calidad

Además al sistema se le realizaron pruebas desde el punto de vista de hardware, ya que la aplicación fue probada en distintos ordenadores, uno de ellos contaba con un procesador Intel(R) Core(TM) I5 CPU 650 @ 3.20 GHZ 3.19 GHZ, RAM 8.00 GB, HDD 1.5 Tera. Por lo que se pudo llegar a la conclusión observando a simple vista, que el sistema a medida que fuese utilizado en una máquina que cuente con altos requisitos no funcionales de hardware , más rápido realizará el proceso de evaluación de la calidad a los videos.

4.8.2 Pruebas Unitarias

Las pruebas unitarias son una forma de probar el correcto funcionamiento de un módulo de código y están dirigidas a analizar clases, métodos y porciones de código que puedan probarse de forma aislada. Estas pruebas validan a través de parámetros de entrada y de salida esperados como trabajan las funciones para cada caso en particular. Si un método varía su funcionamiento en dependencia del contexto deberá realizarse una prueba unitaria por cada caso. El uso de las pruebas unitarias permitirá que: Los errores sean más fáciles de localizar: bastará con ejecutar la batería de “Colecciones de pruebas” y ver qué módulos no las pasan. Los errores están más acotados: cuando un programa falla, muchas veces no se sabe de dónde pueden venir los problemas. Con las pruebas unitarias se consigue acotar los errores, sabiendo qué módulos no están pasando las pruebas. (Dominguez Mora, 2009)

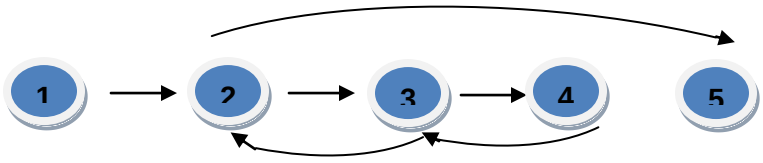
Se reducen los “efectos secundarios”: muchas veces, cuando se quiere arreglar algo bajo presión, se cometen otros errores, o no se tiene en cuenta ciertos aspectos, que hacen que el programa deje de funcionar por otro lado. Incluso a veces, es más peligroso arreglar un error que dejarlo como está, ya que se puede subsanar el error, pero generar otros distintos. Al aplicar las pruebas unitarias es más fácil controlar el programa pues se asegura que todo funciona tal y como se esperaba. Se da más seguridad al programador: normalmente, la persona que ha programado un módulo no es la misma que la que debe corregir sus errores. Esto crea una sensación de inseguridad al programador pues a la hora de corregir un error, no tiene la certeza de que su corrección no va a afectar a otros módulos que desconoce. Las pruebas unitarias aseguran que una corrección no repercuta en otros módulos y permite al programador centrarse en su trabajo. (Dominguez Mora, 2009).

A continuación se exponen algunos métodos analizados manualmente usando pruebas unitarias.

```
double CImagen::Calcular_DMedia(CvMat *matriz)
{
    CvScalar Valor_Matriz;
    double Valor_Media;

    for(int i=0;i<matriz->width;i++)
    {
        for(int j=0;j<matriz->width;j++)
        {
            Valor_Matriz = cvGet2D(matriz, i, j);
            Valor_Media += Valor_Matriz.val[0];
        }
    }

    return Valor_Media/(matriz->height * matriz->width);
}
```



Paso 1: Generar Grafo de flujo de datos.

Paso 2: Cálculo de complejidad ciclomática

V(G)= Número de Aristas - Número de Nodos + 2

V(G)= 6- 5+2= 3

Paso 3: Caminos básicos

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

CB2: 1,2,5

CB3: 1,2,3,2,5

Paso 4: Caso de prueba para el camino básico.

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

V(G)= 6- 5+2= 3

Caso de prueba: Calcular_DMedia

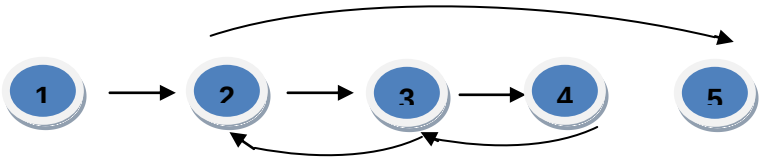
Entrada: Una matriz,CvMat *matriz

Resultado esperado: Devuelve el promedio de los valores de la matriz que pasa por parámetros.

Resultado de la Prueba: Satisfactorio

```
CvMat* CImagen::Calcular_Diferencia(CvMat *mat_orig, CvMat *mat_resul)
{
    CvScalar scala_o;
    CvScalar scala_r;
    CvScalar resta;

    for(int i=0;i<mat_orig->height;i++)
    {
        for(int j=0;j<mat_orig->width;j++)
        {
            scala_o=cvGet2D(mat_orig,i,j);
            scala_r=cvGet2D(mat_resul,i,j);
            resta.val[0]=scala_o.val[0]-scala_r.val[0];
            cvSet2D(mat_orig,i,j,resta);
        }
    }
    return mat_orig;
}
```



Paso 1: Generar Grafo de flujo de datos.

Paso 2: Cálculo de complejidad ciclomática

V(G)= Número de Aristas - Número de Nodos + 2

V(G)= 6- 5+2= 3

Paso 3: Caminos básicos

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

CB2: 1,2,5

CB3: 1,2,3,2,5

Paso 4: Caso de prueba para el camino básico.

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

V(G)= 6- 5+2= 3

Caso de prueba: Calcular Diferencia

Entrada: Dos matrices, CvMat *mat_orig, CvMat *mat_resul

Resultado esperado: Calcula la diferencia entre las matrices que son pasadas por parámetros.

Resultado de la Prueba: Satisfactorio

El resultado de las pruebas realizadas al software fue satisfactorio, se comprobó que las respuestas del sistema son las esperadas y que el mismo cumple con las especificidades planteadas y que a su vez coinciden con las descripciones de los casos de usos planteados con anterioridad. Con las pruebas aplicadas se comprobó que el sistema cumple con lo objetivos propuesto.

4.9 Validación de la solución propuesta.

En este epígrafe se demuestra la veracidad de la aplicación Digital Video Quality (DVQ). Para comprobar que los resultados de esta son los correctos se toman un grupo de secuencias de video, las cuales se escogen teniendo en cuenta los formatos estipulados por la plataforma. A estas secuencias se le determina la calidad mediante el Software MSU Video Quality Measurement Tool del grupo de video MSU Graphics & Media Lab de la Universidad Estatal de Moscú. Aunque MSU no implementa la métrica Watson Modificada, si cuenta con VQM, la cual es similar a Watson Modificada, ya que es una de las métricas en las que se basa. Por otra parte se realiza el mismo procedimiento pero ahora mediante el software que fue implementado como respuesta al problema de investigación, Digital Video Quality (DVQ). Teniendo los resultados de las dos aplicaciones entonces se realiza una comparación, buscando que tan similiares son los valores obtenidos de ambos. Es importante decir que VQM considera que los

valores obtenidos de la evaluación entre (0- 0.80) son buenos, (0.81-1.70) son regulares, (1.71-3.0) son malos y mayores que 3 los consideras muy malos.


Sec_Captura1


Sec_Entrevista


Sec_Serie1


Sec_VideoClip1


Sec_Serie2


Sec_VideoClip2


Sec_VideoClip3


Sec_Captura2

A continuación se detallan los valores obtenidos para cada una de las secuencias:

Secuencias de evaluación	VQM	Clasificación	Watson Modificada	Clasificación
Sec_Captura1	2.42883	Mala	2.16006	Mala
Sec_Captura1(h264)	0.91613	Regular	1,26175	Regular
Sec_Captura1(xvid)	1.99363	Mala	2,32667	Mala
Sec_Entrevista(h264)	5.25696	Muy Mala	5. 25696	Muy Mala
Sec_Entrevista(xvid)	0.94335	Regular	0.75335	Regular
Sec_Serie1	0.35114	Buena	0.55621	Buena
Sec_VideoClip1	0.95786	Regular	1.15876	Regular
Sec_Serie2	3.95844	Muy Mala	4.65841	Muy Mala

Sec_VideoClip2	1.58425	Regular	1.52478	Regular
Sec_VideoClip3	0.25487	Buena	0.52451	Buena
Sec_Captura2	1.25462	Regular	1.04585	Regular

Tabla 4: Comparación DVQ vs VQM

Los resultados obtenidos demuestran la validez del software Digital Video Quality, ya que los valores alcanzados son muy similares a los arrojados en la evaluación mediante el software MSU. Aunque los valores en la mayoría de las secuencias no son los mismos, debido a que la métrica Watson Modificada no tiene en cuenta algunos parámetros importantes para VQM, si en los casos analizados quedaron en el mismo rango de clasificaciones.

4.10 Conclusiones

En este capítulo se pudo observar el resultado del análisis de los diferentes diagramas y principios ilustrados durante el período de construcción de la solución que se propone. Se realizaron los diagramas de clases del diseño, permitiendo el inicio de la implementación de la aplicación.Se escogió como patrón arquitectónico el capas específicamente el tres capas, lo cual representa la arquitectura escogida para el desarrollo del sistema y un mejor funcionamiento por parte del mismo. Todo esto permitió la comprensión de los flujos de los Casos de Uso del Sistema, lo que facilita la implementación de estos. Quedaron descritos los principios de diseño para el sistema propuesto, definiendo además el diagrama de despliegue y el diagrama de componentes. Además se realizaron pruebas al sistema como una forma de validar correctamente lo que se propuso como alcance del trabajo. Arrojando como resultado que la aplicación cumple de manera satisfactoria, con las funcionalidades identificadas durante el proceso de desarrollo del software.

Conclusiones Generales

Una vez culminado el desarrollo de la investigación del presente trabajo de diploma, la implementación del sistema y validación de la aplicación a través de las pruebas, es posible arribar a las siguientes conclusiones:

- El sistema desarrollado para llevar a cabo el proceso de evaluación de la calidad de los videos convertidos en la plataforma PTARTV, basado en métricas de detección de errores recibe como nombre Digital Video Quality (DVQ). El mismo tiene una gran importancia en el campo del procesamiento de video digital, ya sea conversión o compresión de video. Pues permite evaluar la calidad de los videos, especificando el índice de calidad de estos, por tanto la aplicación desarrollada puede llegar a ser de de gran utilidad en esta rama. Para ello es posible afirmar que se ha dado cumplimiento al objetivo general trazado desde inicio del trabajo.

- Con el empleo de los métodos de investigación teóricos y empíricos se logró conocer el estado del objeto de estudio de la investigación. Analizando los principales conceptos asociados al tema de investigación, comprendiendo de manera satisfactoria toda la situación problemática.

- Se realizó el estudio de otras aplicaciones existentes dirigidas a la evaluación de calidad en los videos. Arrojando como resultado que son muy pocas las soluciones existentes en esta rama, pues las que existen son patrocinadas por países desarrollados. Para hacer uso de las mismas se hace engorroso su utilización, ya que los países subdesarrollados deben pagar altos precios. Además no son adaptables ni configurables al flujo de trabajo de la plataforma PTARTV.

- Se obtuvo una valoración de las tendencias y tecnologías actuales que permitieran la correcta selección de entorno de desarrollo integrado, framework, lenguajes de programación, gestor de base de datos, librerías, metodologías de desarrollo y herramienta CASE.

- El proceso de desarrollo del software realizado permitió generar todos los artefactos y documentación correspondiente al mismo, lo que sentó las bases para el desarrollo del producto final.

- La implementación del algoritmo DVQ contribuyó a elevar el nivel de calidad de las transmisiones televisivas, ya que se logró desarrollar el producto final que evalúa la calidad en los videos de la plataforma PTARTV. Permitiendo además que los videos no sean transmitidos con errores y evitando causar molestias en los televidentes.

- El trabajo realizado tributa a la plataforma PTARTV como experiencia en el procesamiento de imagen y video digital.

- La solución implementada agiliza y automatiza la evaluación de la calidad en los videos de la plataforma PTARTV.

- La utilización de la solución informática DVQ en una televisora, posibilita disminuir el número de mano de obra y materiales para el chequeo de la calidad en los videos.

- El presente trabajo forma parte de una excelente propuesta documentada que le permitirá a la universidad seguir ampliando su desarrollo tanto en proyectos como aplicaciones que permitan evaluar calidad de los videos.

Recomendaciones

- Se recomienda comenzar a utilizar el sistema que permita automatizar la evaluación de la calidad en los videos de la plataforma PTARTV.

- Continuar con el estudio y análisis del resto de los procesos que tienen lugar en la evaluación de la calidad de los videos con el objetivo de ampliar las funcionalidades del sistema propuesto.

- Seguir investigando en función de optimizar los resultados obtenidos y robustecer la solución propuesta.

- Realizar otros tipos de pruebas que permitan verificar la calidad del sistema.

- Realizar un manual de usuario para un mejor entendimiento del sistema, por parte del usuario.

Referencias Bibliográficas

Díaz Castro, Maylé y Rivera López, Johanny. 2007. Definición del sistema de información y control de los grupos electrógenos de emergencia. La Habana : s.n., 2007.

Casas Hernández, Pedro, Guerra Vidal, Diego y Irigaray Bayarres, Ignacio . 2005. Calidad de Servicio Percibida en Servicios de Voz y Video sobre IP. 2005.

Microsoft. MSND. [En línea] Microsoft Corporation. [Citado el: 10 de 11 de 1.] Http://msdn.microsoft.com/es-es/library/at62haz6%28VS.80%29.aspx.

Rivera Correa , Olga María y Navarro Sánchez , Arlen. 2010. Sistema para la Gestión de Información referente a Pegaduras de Tuberías durante la perforación de pozos petroleros.SISGIP. La Habana : s.n., 2010.

Aguilar, Ana Rosa Gallego. 2006. Modelos visuales en el análisis de la calidad de la imagen . Madrid : Universidad Politécnica De Madrid, 2006.

Álvarez, Diego Sarasúa. 2010. Aproximación al análisis de secuencia de video codificadas en H.264. Madrid : Universidad Autónoma De Madrid, 2010.

Alvarez, Elaine Morales. 2010. Módulo de Gestión de Errores de la Plataforma de Televisión Informativa PRIMICIA . UCI, Ciudad de la Habana : s.n., 2010.

Andres Gonzalez, Carlos de Jesus. 2010. Diseño del subsistema de configuración de la Plataforma de Televisión Informativa, PRIMICIA. 2010.

Antonio Foncubierta Rodríguez, Ramón Cerquides Bueno. Análisis de calidad de video H.264 en streaming sobre HSUPA. S.l. : Dpto. De Teoría de la Señal Universidad de Sevilla.

Ayala, Dayamí Chávez. 2010. Trabajo de Diploma Arquitectura de la Plataforma de Transmisión Abierta para Radio y Televisión. La Habana : s.n., 2010.

Bautista, Jose M. Programación Extrema. S.l. : Universidad Unión Bolivariana.

2010. Blog – Ramip. [En línea] 18 de 8 de 2010. [Citado el: 25 de 11 de 2010.] Http://www.ramip.net/lenguajes-de-programacion/que-son-los-lenguajes-de-programacion.html.

Booch, Grady, Rumbaugh, James y Jacobson, Ivar. El Lenguaje Unificado de Modelado. Madrid : s.n.

Buján, Leitniz Perez. 2010. VTK. La Habana : s.n., 2010.

Campos, Armando Mora. Octubre del 2008. Tesis Doctoral 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, España : s.n., Octubre del 2008.

Carlos Esteban Baz Hormigos.Septiembre,2009. Medidas de Calidad Subjetiva en Secuencias de Video. Madrid, España : s.n., septiembre,2009.

Córdova, Eridniel Suárez Contreras. Orestes Lima. 2008. Diseño de una Plataforma de Transmisión Abierta para la Radio y la Televisión. . La Habana : s.n., 2008.

Corporación IBM. 2006. Ayuda de RUP. 2006.

Dominguez Mora, Dunier. 2009. Implementación de los subsitemas Web y Transferencia de la Plataforma de Transmisión Abierta para Radio y Televisión. 2009.

Gómez, Alberto Gómez y De Abajo Martínez, Nicolás. Los sistemas de información de la empresa.

González, Carlos de Jesús Andrés. 2010. Trabajo de Diploma Diseño del Subsistema de Configuración de la Plataforma de Televisión Informativa, PRIMICIA . La Habana : s.n., 2010.

Jacobson, Ivar , Booch, Grady y Rumbaugh, James. 2000. El Proceso Unificado de Desarrollo de Software. Madrid : Adisson Wesley, 2000.

Kiware. 2009. Visualization Toolkit. [En línea] Kitware, 2009. [Citado el: 15 de 2 de 2011.] Http://www.vtk.org.

Larman, Craig. UML y Patrones.

Maldonado, Enrique Almeida. 2009. Trabajo de Diploma Sistema de Catalogación de Medias. La Habana : s.n., 2009.

Margaret Pinson, Stephen Wolf ,Phillip G. Austin ,Andrea Allhands. 2000-2002. Video Quality Measurement PC .User's Manual . 2000-2002.

Mora, Sergio Luján. C++ paso a paso.

Nokia Corporation. 2011. QT. [En línea] 2011. [Citado el: 5 de 2 de 2011.]

Pressman, Roger S. 2002. Ingeniería de Software, un enfoque práctico. Quinta Edición. S.l. : Mc Graw Hill, 2002.

Project, Compression. 2010. Video MSU Herramienta de medición de calidad. [En línea] 2010.

Sánchez, Esmeralda Guindel. 2009. Calidad y seguridad de la información y auditoría informática. Madrid : Universidad Carlos III De Madrid, 2009.

Software, Departamento Central de Ingenieria de. 2010. Introducción a RUP y UML. UCI,La Habana : s.n., 2010.

Somerville, Ian. 2005. Ingeniería de software 7ma Edición. España : s.n., 2005.

Glosario de Términos

Píxel: (*acrónimo del* inglés *picture element*, "elemento de imagen") es la menor unidad homogénea en color que forma parte de una imagen digital, ya sea esta una fotografía, un fotograma de video o un gráfico.

Media: película, imagen o cualquier otro material audio visual que requiere de un uso especial de equipamiento para visualizarlo.

Servidor de medias: servidor donde se van a almacenar los archivos multimedia con el formato requerido.

Digitalización: proceso de cuantificar una señal analógica, convertirla y representarla en forma digital.

Codificación de Video: proceso de conversión de video de un formato a otro usando códec.

Compresión de Video: es una forma de compresión de datos diseñada para reducir el tamaño de los archivos de video.

Caso de Uso: secuencias de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de las secuencias.

Prototipo: maqueta visual funcional o no de la futura aplicación. Este puede ser una imagen o una aplicación software que simule funcionalidades del software.

IEEE: corresponde a las siglas de (*The Institute of Electrical and Electronics Engineers)*, el Instituto de Ingenieros Eléctricos y Electrónicos, una asociación técnico-profesional mundial dedicada a la estandarización.

Protocolo: Conjunto de normas y procedimientos útiles para la transmisión de datos, conocido por el emisor y el receptor.

Servidor: Una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes.

Vectores de movimiento: Es una estimación del desplazamiento horizontal y vertical de cada región de una cierta imagen con respecto a uno o varios frames de la misma secuencia.

Luminancia: Es la densidad angular y superficial de flujo luminoso que incide, atraviesa o emerge de una superficie siguiendo una dirección determinada.

Crominancia: Es el componente de la señal de video que contiene las informaciones del color.