***Universidad de las Ciencias Informáticas***
***Facultad 6***


# ***Título:*** *Herramienta de Monitoreo de Señales Digitales para la Plataforma de Transmisión Abierta para Radio y Televisión.*

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

***Autor:*** *Rafael Simón Martín*

***Tutor:*** *Ing. Abel Díaz Berenguer*

***Co-Tutor:*** *Ing. Neysis Hernández Díaz*

***La Habana, junio 22 de 2011.***
***"Año 53 de la Revolución"***

## ***Dedicatoria***

***Este trabajo está dedicado a todos aquellos que han luchado para que los jóvenes del presente tengamos la posibilidad de estudiar de manera gratuita y poder ser buenos profesionales.***

***A mi madre por estar cada vez que la necesito.***

***A mi hijo por darme la fuerza y reforzar mi espíritu de lucha.***

## ***Agradecimiento.***

***Agradezco a mi madre, por apoyarme, aconsejarme y ser mi principal inspiración en el estudio.***

***A mi hijo, por ser el ángel que me obliga a superarme para que pueda tenerme como guía en el estudio y en la preparación intelectual.***

***A mi hermana, que su inteligencia en el estudio me ha ayudado a tratar de siempre ser como ella.***

***A mi novia, por ayudarme en el estudio durante mi estancia en la universidad y ser una de las mejor cosas que me ha pasado en la vida.***

***A la Revolución Cubana por darme la maravillosa posibilidad de ser un profesional.***

***A mis compañeros de la universidad.***

***A los todos los profesores que he tenido, y que de una manera u otra me han ayudado y formado para poder ser un buen profesional.***

***A mi tutor, que en todo momento me ha apoyado, además de aportar gran parte de sus conocimientos para que este trabajo llegara a ser posible.***

# ***Declaración de Autoría.***

Declaro que soy el único autor del trabajo titulado:

Herramientas de monitoreo de las Señales Digitales para la plataforma abierta de radio y televisión.

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

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

_____ _____

Rafael Simón Martín Ing. Abel Díaz Berenguer

## ***Datos de contacto.***

Tutor(a):

Nombre y apellidos: Abel Díaz Berenguer.

Sexo: X M F

Institución: Universidad de las Ciencias Informáticas.

Dirección de la institución: Carretera a San Antonio de los Baños, Km. 2 ½,

Reparto: Torrens.

Municipio: Boyeros.

Provincia: Ciudad de La Habana.

Correo electrónico: aberenguer@uci.cu.

Teléfono del trabajo:-

Teléfono particular:-

Cargo del trabajador: Profesor

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

Año de graduación: 2009.

Institución donde se graduó: Universidad de las Ciencias Informáticas.

## ***Resumen.***

A través de los años y luego de investigar vías más fáciles para difundir cualquier tipo de acontecimiento, ya sea informativo o recreativo, se comienza a pensar en una herramienta que cumpliera estas expectativas y es cuando comienza a surgir la televisión. La televisión desde sus inicios comenzó a observarse por un pequeño sector de la sociedad ya que era algo muy costoso. En nuestros días es algo necesario para muchos, porque mediante la misma podemos lograr informarnos, recrearnos, además de aprender e instruirnos acerca de diversos temas.

A pesar del gozo que ofrece este beneficio en los hogares y en algunas instituciones que lo requieran, muchas veces se desconoce el esfuerzo y el trabajo que hay detrás de esto. Diversas personas desempeñan todos los días varios roles para hacer posible que millones de personas logren ver algo verdaderamente satisfactorio. Dentro de todo este inmenso mundo se encuentra el proceso de monitorear las señales de televisión.

En este documento se presentan los resultados de la investigación y el proceso de desarrollo realizado durante la construcción de la herramienta para monitorear las señales transmitidas en la Dirección de Televisión Universitaria perteneciente a la Universidad de las Ciencias Informáticas. Además se determinan los principales conceptos asociados a la investigación y las herramientas de desarrollo necesarias para la construcción de la herramienta.

## ***Palabras claves.***

- Transmisión.
- Señal.
- Monitoreo.
- Herramienta.
- Sistema Digital.

# ***Índice de tablas.***

# ***Índice de figuras.***

# *Índice.*

# *Introducción.*

Con el desarrollo de las tecnologías de la información y las comunicaciones se le proporcionó al mundo una amplia gama de servicios, por ejemplo: aplicaciones, tecnologías, equipos y programas informáticos que le han dado a las personas prestaciones que muchas veces agilizan su trabajo. Además ofrecen facilidades de búsqueda y recuperación de información creando los canales de comunicación necesarios para esto. Estos servicios ofrecen diferentes beneficios como son rápida comunicación entre las personas, se ha podido visualizar en tiempo real lo que sucede en otras partes del mundo y se han creado sistemas computarizados que facilitan el trabajo, como por ejemplo la telefonía e Internet, los televisores, los ordenadores y las redes que son necesarios para el uso de las mencionadas tecnologías.

Desde las últimas décadas del siglo XX la televisión se convirtió en bandera tecnológica, con un gran auge debido a la necesidad de información y entretenimiento de las personas. La televisión ha permitido que la información recorra el mundo a través de avances técnicos que permitieron la trasmisión y el control de las señales; además de poder ser grabadas y editadas por los receptores de las mismas.

Los canales televisivos han constituido un medio de difusión para mantener a las personas informadas debido a la necesidad de comunicarse, expresarse y estar al tanto de lo que ocurre a diario. La televisión en sus inicios comenzó a transmitirse enviando ondas a través de un sistema analógico la cual es vista como televisión analógica. La misma derrocha mucho espectro electromagnético pues los parámetros de las imágenes y el sonido se representan a través de magnitudes analógicas de una señal eléctrica, donde el trasporte de esta señal hasta los hogares ocupa mucho recurso.

Con el paso de los años la televisión se fue perfeccionando, se ha efectuado un proceso de digitalización de la señal analógica. Este proceso es realizado por un convertidor analógico-digital que permite someter la señal de televisión a procesos muy complejos, sin degradación de calidad, obteniendo una señal digital, esto ofrece múltiples ventajas. Las señales digitales permiten recibir audio y video con imágenes de alta resolución, esto no es producto de casualidad sino de más de 50 años de evolución de la televisión analógica.

El desarrollo de la televisión digital es posible debido a las señales digitales que son un tipo de señales generadas por algún fenómeno electromagnético**.** Se ha demostrado que las mismas ofrecen mayor inmunidad al ruido, permiten detectar y corregir errores, además aseguran un menor consumo de energía y ofrecen la posibilidad de encriptar la información con técnicas de compresión de datos.

Quizás no sea muy fácil medir la calidad de las señales digitales, debido a su naturaleza ya que el espectro en frecuencia de una señal digital no revela los atributos de la señal. Sin embargo se puede procesar una señal para obtener una disminución del nivel de ruido y para mejorar la presencia de determinados matices, como los graves o los agudos. Realizando un monitoreo de estas señales se podrá llevar un control visual de las mismas en cuanto a imagen y audio, así como medir e informar aspectos asociados al funcionamiento de los servicios de videos que se están trasmitiendo.

Durante el monitoreo continuo de las señales digitales se provee una fácil visualización del contenido como videos en miniaturas que se pueden ver a través de una interfaz de escritorio. También, para estar informado de una forma más detallada, se puede seleccionar un servicio en transmisión y visualizar sus parámetros.

Cuba no ha estado ajena a estas transformaciones en el mundo de las tecnologías de la información y principalmente referente al desarrollo de la televisión. Debido a esto se creó en septiembre del 2002 la Universidad de las Ciencias Informáticas (UCI) también conocida como la ciudad digital. La misma está estructurada en siete facultades con sus propias líneas de investigación y desarrollo. Dicha institución está encaminada a promover la producción del software en el país. La Facultad 6 de la UCI está desarrollando proyectos orientados a servicios televisivos para extenderlos hacia toda la comunidad universitaria.

En esta facultad se encuentra el centro de desarrollo GEYSED que cuenta con los departamentos de Geoinformática y Señales Digitales. El departamento de Señales Digitales asume la realización de la Plataforma de Transmisión Abierta para Radio y Televisión (PTARTV) que cuenta con varios subsistemas: Transferencia, Gestión de Medias, Web, Programación, Seguridad, Reporte, Transmisión y Administración de la Transmisión. Los dos últimos actualmente se encuentran en un proceso de personalización. PTARTV constituye una solución informática para la Dirección de Televisión Universitaria de la UCI.

Los desarrolladores de PTARTV han detectado que el proceso de transmisión de contenidos multimedia se realiza de forma tal que es necesario un elevado número de especialistas. Con el inconveniente de realizar este proceso de forma prácticamente manual, lo que trae consigo pérdida de tiempo productivo, a lo que se puede adicionar la posibilidad de errores humanos que pueden hacer colapsar todo el proceso.

Entre las funcionalidades que pertenecen al subsistema Administración de la Transmisión de PTARTV se encuentra: monitorear las señales digitales. Esta funcionalidad posibilita la visualización de los canales que se transmiten en busca de posibles errores además del control visual de audio y video. En la DTU existen serios problemas para el monitoreo de las señales que se encuentran en transmisión, debido a que no existe una herramienta que permita monitorear las señales digitales y poder visualizar lo que se transmite en tiempo real. Además no se puede visualizar todos los canales a la vez y si existe un error no hay manera de notificarlo o de avisar mediante un sistema de alarma.

Por este motivo se propone realizar la investigación a partir del siguiente **problema a resolver**:

- ¿Cómo monitorear las señales que se difunden en la Plataforma de Transmisión Abierta para Radio y Televisión?

**El objeto de estudio** de este trabajo lo constituye: los procesos asociados al monitoreo de señales digitales y el **campo de acción** es: automatización de los procesos de monitoreo de señales que se difunden en la PTARTV.

Para dar solución al problema planteado se propone como **objetivo general**:

- Desarrollar una herramienta que permita el monitoreo de las señales que se transmiten en la Plataforma de Transmisión Abierta para Radio y Televisión.

Teniendo en cuenta lo anteriormente planteado se propone como **idea a defender**: el desarrollo de una aplicación para el monitoreo de las señales digitales difundidas por la Plataforma de transmisión abierta para radio y televisión se garantizará controlar las transmisiones.

Se proponen las siguientes **tareas de investigación** para darle solución al objetivo general:

1. Caracterizar los mecanismos de monitoreo de señales en la Dirección de Televisión Universitaria (DTU).
2. Caracterizar aplicaciones o soluciones existentes que permitan analizar el funcionamiento de sistemas para el monitoreo de señales digitales.
3. Determinar las tendencias y tecnologías actuales más factibles para el desarrollo de la aplicación.
4. Desarrollar los artefactos y documentación correspondientes al proceso de desarrollo del software.
5. Validar la herramienta desarrollada.

Para dar cumplimiento a las tareas planteadas se utilizan diferentes métodos científicos, los cuales se muestran a continuación.

- **Métodos teóricos:**
  - **Analítico-Sintético:** Se utiliza este método para realizar una búsqueda bibliográfica referente a este tema y seleccionar las más adecuada para determinar la esencia del problema, relacionada con los elementos esenciales que implican la existencia de dificultades, carencias o limitaciones en cuanto al monitoreo de las señales digitales, así como establecer sus fundamentos, el diseño de soluciones y el análisis del impacto de las mismas.
  - **Histórico-Lógico:** Se utiliza este método con el fin de conocer los antecedentes, etapas significativas, funcionamiento de los sistemas de monitoreo de las señales digitales. Además con el conocimiento profundo de la esencia de los sistemas de monitoreo y señales digitales se analiza la lógica interna de su desarrollo y se expresa la esencia del objeto de estudio.
  - **Inductivo-Deductivo:** Para realizar el análisis e inferencia respectiva sobre el estudio realizado en el objeto y campo lo que permitirá arribar a la propuesta que posibilite solucionar el problema y las conclusiones correspondientes.

- **Métodos empíricos**:

- **Observación:** Se utiliza en la realización del estudio preliminar del entorno y para la recopilación de la información necesaria. Durante la Observación, se realizaron visitas al centro GEYSED para observar los procesos, lográndose clasificar y entender la situación problemática.
- **Entrevistas:** Se utiliza a los estudiantes del centro GEYSED que pertenecen al departamento de Señales Digitales, con la finalidad de corroborar la necesidad de un mecanismo que garantice el control de las señales, teniendo en cuenta sus opiniones sobre qué elementos deben incluirse.
- **Encuestas:** Se realiza a estudiantes utilizada como pretest y postest: Se aplican dos encuestas con el objetivo de constatar, en conjunto con los demás miembros del proyecto para obtener información acerca de las experiencias hasta el momento del tratamiento que se le da a las señales digitales con el fin de valorar la factibilidad y efectividad de una propuesta de control y seguimiento.
- **Criterio de especialistas**: Se consultan especialistas de la DTU para valorar la importancia, novedad, necesidad, aplicabilidad y efectividad de una herramienta de monitoreo de señales digitales.

Para esto se ha seleccionado una población de 15 personas entre los cuales se encuentran estudiantes y profesores del centro así como especialistas de la DTU. Para el desarrollo de las entrevistas y encuestas se toma como muestra 2 profesores y 2 estudiantes, además de 1 trabajador de la DTU. La muestra representa el 30 por ciento de la población la cual fue seleccionada de forma intencional y aleatoria empleando técnica de muestreo no probabilístico.

Para una mayor comprensión del trabajo de diploma el mismo se encuentra estructurado de la siguiente forma:

**Capítulo 1:** Se encuentran los temas relacionados con el objeto de estudio, descripción del entorno donde se encuentra el negocio y la organización, así como la descripción detallada de la situación problemática y el análisis de soluciones similares que puedan brindar respuesta al problema planteado.

**Capítulo 2:** Abarca un estudio y valoración sobre las tendencias y tecnologías actuales que van a ser utilizadas en el desarrollo de la aplicación. Se realizan

comparaciones y se seleccionan las más factibles para el desarrollo del sistema propuesto.

**Capítulo 3:** Se propone la solución al problema planteado teniendo en cuenta la descripción de los procesos del negocio, así como los requisitos que el sistema debe cumplir y la identificación de los casos de uso del sistema con su respectiva descripción y representación.

**Capítulo 4:** Se construye la solución propuesta al desarrollo de la aplicación pero ya en su sentido más completo, así como el análisis y diseño del sistema que incluye todos los diagramas necesarios así como y sus respectivos diagramas de clase, así como los modelos de “Implementación” y de “Despliegue” para completar la modelación del sistema.

**Capítulo 5:** Se valida la herramienta desarrollada mediante los tipos de pruebas de funcionalidad y de rendimiento, incluye ejemplos de dichas pruebas así como los resultados de las mismas.

Además se incluyen las conclusiones, recomendaciones, referencias bibliográficas, glosario de términos.

# *Capítulo 1: Fundamentación Teórica.*

## *1. Introducción.*

En este capítulo se abordarán los elementos relacionados con la fundamentación teórica que incluye la definición de los conceptos asociados al dominio del problema, así como la descripción detalla de la situación problemática, además del estudio de las soluciones o sistemas de monitoreo de señales digitales existentes a nivel nacional e internacional.

## *1.1. Conceptos asociados al dominio del problema.*

### *1.1.2. Herramienta.*

Con el paso del tiempo el hombre se ha enmarcado en la era del desarrollo y ha buscado vías que permitan realizar su trabajo manual de una forma más sofisticada y con el menor esfuerzo físico posible. Se puede llegar a la conclusión que una herramienta, no es más que un instrumento mediante el cual se realizan tareas de una manera más fácil y cómoda. Las personas usan las herramientas con el fin de facilitar las acciones de sus trabajos necesitando el menor número de personas implicadas en el desarrollo del mismo y con mayor calidad y eficiencia. (1)

### *1.1.3. Monitoreo.*

El origen de la palabra monitoreo surge a partir de la palabra monitor que no es más que el aparato que toma imágenes de instalaciones filmadoras o sensores y que permite visualizar algo en una pantalla. Por lo tanto un monitor posibilita controlar en una pantalla una situación determinada.

Según el libro “Monitoreo e indicadores” representado por La Organización de Estados Iberoamericanos, en Guatemala [La teoría de la planificación del desarrollo define el seguimiento o monitoreo como un ejercicio destinado a identificar de manera sistemática la calidad del desempeño de un sistema, subsistema o proceso a efecto de introducir los ajustes o cambios pertinentes y oportunos para el logro de sus resultados y efectos en el entorno. Así, el monitoreo permite analizar el avance y proponer acciones a tomar para lograr los objetivos; Identificar los éxitos o fracasos reales o potenciales lo antes posible y hacer ajustes oportunos a la ejecución]. (2)

El monitoreo consiste en dar seguimiento a una acción, es decir: supervisión o control a través de un monitor donde se visualizara una serie de acontecimientos o acciones que se están efectuando en un instante de tiempo determinado, controlando de esta manera uno o más parámetros con el objetivo de detectar cualquier anomalía.

### *1.1.4. Señal.*

Señal es un término que proviene del latín signālis. Se trata de un signo, seña, marca o medio que informa, avisa o advierte algo. Una señal también puede ser un gesto que realiza una persona para advertir a otra una determinada circunstancia. Este tipo de señal puede ser hecha mediante un movimiento de manos o de los brazos. (1)

Existen diferentes tipos de señales como por ejemplo: (3)

- **Las señales de seguridad**: La seguridad aparece a partir de que existe cualquier tipo de peligro y la manera de avisar de que el mismo existe, es mediante el uso de símbolos que representen la presencia de la misma. La prevención de que el peligro está cerca es importante para la preservación de nuestras vidas.
- **Las señales de tránsito**: Son las que mientras transitemos mediante el uso de vehículos u otro medio de transporte nos indiquen lo que debemos y no hacer para prevenir accidentes.
- **Las señales analógicas**: Son señales como por ejemplo la luz, el sonido, la energía y otros que tienen una variación continua. Incluso en la descomposición del arcoíris se puede ver que se realiza de una manera suave y continua.
- **Las señales digitales**: Son señales que se generan a partir de algún fenómeno electromagnético, en el que cada signo que codifica el contenido de la misma, puede ser analizado en términos de magnitudes que representan valores discretos, en lugar de valores dentro de un rango determinado. Como por ejemplo el interruptor de la luz, que solo puede tomar valores o de encendido o de apagado.

  Las señales digitales tienen ventajas como por ejemplo:

  - Ante la atenuación, puede ser amplificada y reconstruida al mismo tiempo, gracias a los sistemas de regeneración de señales.

- Cuenta con sistemas de detección y corrección de errores, en la recepción.
- Se ven menos afectadas a causa del ruido ambiental en comparación con las señales analógicas.
- Permite la generación infinita sin pérdida de calidad.

Las señales digitales también tienen algunas desventajas como son:

- Necesita una conversión analógica-digital previa y una decodificación posterior en el momento de la recepción.
- Requiere mayor ancho de banda que las analógicas para ser trasmitida.

De los tipos de señales mencionados anteriormente solo serán objeto de estudio las señales digitales y analógicas.

Una señal puede ser también la variación de una corriente eléctrica u otra magnitud física que se utiliza para transmitir información. Por ejemplo, en telefonía existen diferentes señales, que consisten en un tono continuo o intermitente, en una frecuencia característica, que permite conocer al usuario en qué situación se encuentra la llamada. Es definida como cualquier cantidad física que varía en el tiempo y que lleva información, generalmente acerca del estado o comportamiento de un sistema, como por ejemplo: radar, música, voz, sonar. (4)

Una vez expuesto los conceptos anteriores se puede resumir que las señales se utilizan para trasmitir información a través de la variación de la corriente eléctrica o de otra magnitud.

### *1.1.5. Sistema digital.*

Son sistemas que utilizan una lógica de dos estados que se representan por dos niveles de tensión electrónica: alto y bajo. A modo de abstracción dichos estados se sustituyen por unos y ceros facilitando de esta manera la aplicación lógica y aritmética (Figura. 1).

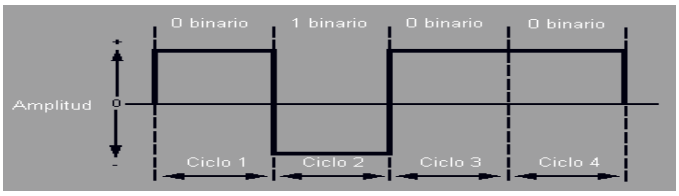



***Figura. 1: Representación de un sistema digital.***

El sistema binario que desarrolla estos sistemas permite almacenar, procesar y transmitir cualquier tipo de información. Se puede hablar de dos tipos de sistemas digitales:

- Sistemas digitales combinacionales: Cuyas salidas dependen del estado de las entradas en un momento dado.
- Sistemas digitales secuenciales: Donde las salidas dependen del estado del momento dado, pero también de estados previos.

El objetivo principal de cualquier sistema de comunicación de datos es ser capaz de tomar el mensaje generado por el emisor, transmitirlo por la línea y que llegue al receptor de la forma más parecida a como fue emitido, es decir, con el menor número de errores posible. Esta regla es la que tratan de seguir tanto los sistemas digitales como los analógicos, no obstante existen algunas diferencias entre ellos:

1) Los sistemas analógicos son más sencillos de implementar porque se actúa menos sobre la señal que se quiere transmitir.
2) Los sistemas digitales, realizan labores de compactación y encriptación de la información, lo que hace que sea necesaria más circuitería en el caso de las comunicaciones digitales.
3) En las comunicaciones digitales se trata de escoger formas para las señales que se adecuen al medio por donde van a ser transmitidas, lo que hace que el efecto del ruido durante la transmisión sea mucho menor. (5)

Resumiendo, a pesar de que las comunicaciones analógicas son más sencillas, las digitales proporcionan una calidad y fiabilidad mucho mayor. Su único inconveniente es la complejidad de los equipos, pero a medida que pasa el tiempo, los costes se reducen y su implantación en todo el ámbito de las comunicaciones es imparable.

## *1.1.6. Transmisión.*

Según la Real Academia Española transmitir no es más que trasladar, transferir, hacer llegar a alguien mensajes o noticias, Comunicar información. (3)

La ITU[1] (antes CCITT[2]) en su norma X.15, define la transmisión de datos como la acción de cursar datos, a través de un medio de telecomunicaciones, desde un lugar en que son originados hasta otro en el que son recibidos. Transmisión de información que consiste en el movimiento de información codificada, de un punto a uno o más puntos, mediante señales eléctricas, ópticas, electroópticas o electromagnéticas. (5)

Objetivos de la transmisión de datos (5)

Los principales objetivos que debe satisfacer un sistema de transmisión de datos son:

- Reducir tiempo y esfuerzo.
- Aumentar la velocidad de entrega de la información.
- Reducir costos de operación.
- Aumentar la capacidad de las organizaciones a un costo incremental razonable.
- Aumentar la calidad y cantidad de la información.

Es importante el análisis de las trasmisiones digitales pues serán objetos de estudio en el desarrollo de la investigación. A continuación se describen las ventajas y desventajas de la misma.

### *1.1.6.1. Ventajas de la transmisión digital.*

- La ventaja principal de la transmisión digital es la inmunidad al ruido. Las señales analógicas son más susceptibles que los pulsos digitales a la amplitud no deseada, frecuencia y variaciones de fase. Esto se debe a que con la transmisión digital, no se necesita evaluar estos parámetros, con tanta precisión, como en la transmisión analógica.
- Se prefieren a los pulsos digitales por su mejor procesamiento y multicanalización que las señales analógicas. Los pulsos digitales pueden guardarse fácilmente, mientras que las señales analógicas no pueden. Además, la razón de transmisión de un sistema digital puede cambiarse fácilmente para adaptarse a diferentes ambientes e interfaces con diferentes tipos de equipamiento.

[1]Unión Internacional de Telecomunicaciones.
[2]**Comité Consultivo Internacional Telegráfico y Telefónico** es un organismo de normalización dependiente de la Unión Internacional de Telecomunicaciones, una agencia de las Naciones Unidas. Este organismo ha sido el responsable de normastales como la X.21 y la X.25.

- Los sistemas digitales utilizan la regeneración de señales, en vez de la amplificación de señales, por lo tanto producen un sistema más resistente al ruido que su contra-parte analógica.
- Las señales digitales son más sencillas de medir y evaluar. Por lo tanto, es más fácil comparar el rendimiento de los sistemas digitales con diferentes capacidades de señalización e información, que con los sistemas analógicos comparables.
- Los sistemas digitales están mejor equipados para evaluar un rendimiento de error (Por ejemplo, detección y corrección de errores), que los sistemas analógicos. (6)

### *1.1.6.2. Desventajas de la transmisión digital.*

- La transmisión de las señales analógicas codificadas de manera digital requieren de más ancho de banda para transmitir que la señal analógica.
- Las señales analógicas deben convertirse en códigos digitales, antes de su transmisión, y convertirse nuevamente a analógicas en el receptor.
- La transmisión digital requiere de sincronización precisa, de tiempo, entre los relojes del transmisor y del receptor.
- Los sistemas de transmisión digital son incompatibles con las facilidades analógicas existentes. (6)

En términos informáticos se puede llegar a la conclusión de que la transmisión consiste en el proceso de envío y recepción de datos entre varios ordenadores o dispositivos, logrando de esta manera la comunicación entre el receptor y transmisor.

## *1.2. Objeto de Estudio.*

En el presente trabajo se ha definido como objeto de estudio los procesos asociados al monitoreo de las señales digitales.

### *1.2.2. Descripción General.*

El monitoreo de las señales digitales es fundamental en cualquier institución que lo requiera, es decir, donde existan principalmente varios flujos de videos a la vez. Es de suma importancia garantizar un ambiente adecuado para desarrollar este tipo de sistema de monitoreo, ya que se necesitan medios para garantizar la transmisión de datos, además de capturar y visualizar los videos en cuestión. La utilización de estos

sistemas brinda la posibilidad de controlar las fallas que ocurren durante la transferencia de la información.

Un sistema de monitoreo posee buena calidad si garantiza:

- **Control y visualización del audio**: Es decir que mientras los videos se están reproduciendo exista un control de si el video se está escuchando o no, además de su visualización.
- **Visualización de los videos**: Mientras los videos están en su estado de reproducción y se están transmitiendo en tiempo real observar cualquier dificultad que el mismo pueda tener durante el tiempo de transmisión.
- **Aviso de error**: Cuando los videos de estén visualizando enviar un mensaje que identifique o notifique la existencia de algún problema en caso de que ocurra.
- **Reporte de errores durante la transmisión**: La transmisión durante su proceso de visualización puede presentar algunos errores que se deben almacenar debido a la necesidad de realizar luego un análisis de los mismos.

### *1.2.3. Descripción actual del dominio del problema.*

La Universidad de las Ciencias Informáticas tiene los objetivos definidos, y entre estos la formación de un gran número de profesionales, los mismos tienen como objetivo fundamental impulsar el desarrollo informático y computarizado en todo el territorio nacional. Esto es posible ya que se apoya fundamentalmente durante la preparación de los profesionales en los centros productivos.

La facultad 6 de la UCI y GEYSED como el centro de servicios y soluciones informáticas para el procesamiento de Señales Digitales y la Geoinformática, cuenta entre sus líneas de investigación y desarrollo con la línea de procesamiento, análisis y trasmisión de contenidos audiovisuales para lo cual ha desarrollado de manera aislada varios tipos de sistemas que dan solución a diversos entornos y campos de aplicación para su uso en diversos sectores de la sociedad, ejemplo de ellos son Primicia, Video Web, la Plataforma de Transmisión Abierta para Radio y Televisión, Video Vigilancia y sistemas de Captura y Catalogación de Medias. Los mismos se caracterizan por funcionalidades y un conjunto de componentes bases comunes de un sistema a otro, pero también particularidades propias que los diferencian.

Primicia, Video Web y la Plataforma de Transmisión Abierta para Radio y Televisión(PTARTV), constituyen sistemas multiplataforma, desarrollados con el objetivo de informar de manera rápida y constante a un grupo significativo de personas ubicados dispersamente utilizando recursos mínimos o ya existentes, presentan una arquitectura del tipo cliente-servidor cuyo estilo arquitectónico es modelo vista controlador. Video-Web permite la gestión y transmisión de contenidos multimedia a través de la red utilizando tecnología streaming. Primicia es una solución integral capaz de proveer un canal de televisión para la transmisión automática y constante de informaciones en distintos formatos (textos, imágenes, sonidos y videos) y PTARTV constituye un sistema que posibilita la transmisión abierta para radio y televisión utilizando tecnología streaming.

El Proyecto PTARTV es el encargado de desarrollar un sistema que automatice todos los procesos que se realizan en una televisora. La DTU es la entidad encargada de transmitir a toda la comunidad universitaria, tanto la televisión nacional como los canales internos de la universidad. El principal abastecedor de software de la DTU es el centro GEYSED. Debido a la importancia de esta plataforma se considera que con la creación y puesta en marcha de la misma. La DTU poseerá una herramienta capaz de automatizar un gran número de procesos, que para la producción de TV, hasta el momento es complejo realizarlo. El presente trabajo se enmarca específicamente en el desarrollo de una herramienta de monitoreo de señales digitales que forma parte de la PTARTV.

### *1.2.3.1. Situación Problemática.*

La DTU posee varios departamentos, de todos ellos, el departamento de Programación es el encargado de llevar a cabo el proceso de transmisión de un material audiovisual. Dicho departamento está dividido en tres áreas: Programación, Centro de Gestión de Información Audiovisual (CGIA) y Transmisión.

El proceso comienza por el área de Programación, donde se realiza una revisión de todos los materiales para formar parte del banco de medias de la DTU. Estos se clasifican en aptos o no aptos. Si el material pasa la revisión satisfactoriamente (material apto), el CGIA procede a la catalogación y almacenamiento del mismo. Desde este momento es posible incluir el audiovisual en un programa de transmisión (comúnmente denominado planificación de transmisión). Cuando Programación incluye el material en una de las planificaciones, entonces se transfiere también al

área de Transmisión. Transmisión inicialmente archiva todas las medias en un servidor que es la fuente para la posterior transferencia hacia cada una de las estaciones de trabajo en las que se apoya la transmisión de los canales que posee la DTU (estaciones de transmisión).

La DTU posee una computadora por cada canal de programa que envían hacia red televisiva de la UCI. Al adaptar los subsistemas Transmisión y Administración a las condiciones actuales de transmisión audiovisual de la DTU, se estructura de la siguiente forma: una computadora personal por cada señal que vaya a transmitirse y otra para cumplir la función de administradora del sistema. Dentro de la DTU existe una persona encargada de velar por la recepción de la transmisión de la información que se quiere hacer llegar a la comunidad universitaria ya sean las señales de la televisión cubana, además de las señales internas.

Una vez centralizada todas las señales por medio de un modulador, en un televisor el encargado de revisar esta información para saber si las mismas se están trasmitiendo debe revisar manualmente señal por señal para saber si estas se están trasmitiendo. Actualmente en nuestro país cada canal se monitorea de manera individual, o sea, en el Instituto Cubano de Radio y Televisión (ICRT) se realiza el monitoreo de cada una de las señales de forma individual, puesto que hasta el momento no ha habido la necesidad de monitorear dos o más señales al mismo tiempo.

Sin embargo la DTU debido al volumen de señales que se trasmiten tiene la necesidad de monitorear varios canales televisivos al mismo tiempo, es por esto, que se ha dado a la tarea de desarrollar un sistema informático, el cual permita realizar el monitoreo de forma automática sin la necesidad de una persona que realice este trabajo manualmente. Una vez automatizado este proceso, el reporte de los errores garantizará conocer si es de audio o video.

Para que esta tarea pueda llevarse a cabo, primeramente es necesario que existan señales digitales, las mismas serán recepcionadas por el sistema de monitoreo, el encargado de mostrar las cualidades de video y audio que tienen los materiales, que están siendo televisados por los usuarios. En caso de que exista algún tipo de error mostrar un mensaje o una señal de error y enviársela al transmisor, al cual se le informa que existe algún tipo de problema, que puede ser que no se está mostrando video o que no presenta muestras de sonido.

## *1.3. Análisis de otras soluciones existentes.*

Durante el proceso de investigación se pudo detectar que existen sistemas capaces de realizar actividades y funciones similares a las a las que se necesitan. A continuación se exponen algunos ejemplos para realizar un análisis de sus potencialidades y dificultades.

### *1.3.1. Soluciones integrales similares.*

- **TV LOG**: Es un sistema de registro de audio y video, para grabar, monitorear y supervisar varias señales de televisión simultáneamente. Esta desarrollado bajo plataforma Linux por ser absolutamente estable para operaciones de grabación continua de 24 horas sin detención o corte algún, no hay corrupción del sistema y es libre de virus y hackers, compresión MPEG 4 (H.264). (7)
- **Visual Video-Sphere**: Es un software de vigilancia por video escalable que ofrece manejo avanzado, herramientas de investigación y generación de reportes de errores. (8)
- **Omnicast de Genetec**: Permite administrar fácilmente las alarmas, control de accesos, reportes de seguridad, aplicaciones de análisis de vídeo. (9)

Los sistemas actuales existentes tienen como inconveniente que son aplicaciones privativas con un alto costo, pero de la anteriormente expuestas solo es de monitoreo de televisión el **TV LOG**.

## *1.4. Conclusiones.*

En este capítulo se detallaron las condiciones y problemas actuales que rodean al objeto de estudio enmarcado en este trabajo lo que permitió conocer los procesos de monitoreo y su funcionamiento. A través de los conceptos y definiciones planteadas, se determinaron las condiciones específicas que rodean al problema lo que garantiza el centro de la investigación y las posibles soluciones que le darán respuesta al objetivo general planteado. Al realizarse el estudio de las aplicaciones existentes a nivel internacional y nacional, se evidenció la necesidad de la creación de una herramienta de monitoreo de las señales digitales, pues las existentes a nivel internacional son muy costosas y la universidad no puede adquirir estos medios, además a nivel nacional no existe una herramienta de monitoreo que se ajuste a las necesidades de la DTU.

# *Capítulo 2: Tendencias y tecnologías actuales.*

## *2. Introducción.*

El desarrollo de las tecnologías para la construcción del software ha ido en aumento en estos últimos años, debido al crecimiento desmedido de las Tecnologías de la Información y las Comunicaciones (TIC). Las empresas dedicadas a la construcción de software han optado por utilizar dichas tecnologías con el objetivo de acelerar sus procesos, permitiendo elevar la calidad y la eficiencia de las mismas.

Según Roger Pressman la Ingeniería de Software constituye una tecnología estratificada con enfoque en la calidad que abarca procesos, métodos y herramientas. Durante el desarrollo de un software es necesario aplicar principios de ingeniería, por lo que en el presente capítulo se detallan aspectos sobre diversas tecnologías que pueden ser utilizadas en la construcción del sistema. El análisis se realiza basándose en las características y especificaciones de las mismas. Una vez concluido el estudio se podrá determinar las tecnologías factibles para desarrollar la herramienta de monitoreo de las señales que se emiten en PTARTV.

## *2.1. Metodologías de desarrollo.*

La Metodología de desarrollo de software encierra un conjunto de técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software. A continuación se realiza un estudio de algunas metodologías.

### *2.1.1. Microsoft Solution Framework (MSF).*

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

MSF tiene las siguientes características: (10)

- Escalable: puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.
- Flexible: es utilizada en el ambiente de desarrollo de cualquier cliente.
- Tecnología múltiple: porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

La metodología MSF se adapta a proyectos de cualquier dimensión y de cualquier tecnología, se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el Modelo de Aplicación.

### *2.1.2. Proceso Unificado de Software (RUP).*

RUP es una metodología que guía el desarrollo de software. Constituye una forma disciplinada de asignar tareas y responsabilidades en un proyecto de desarrollo, definiendo quién hace qué, cómo y cuándo. Tiene como objetivo fundamental asegurar la producción de software de calidad dentro de plazos y presupuestos predecibles. (11)

RUP además incorpora el concepto de "mejores prácticas" para la ingeniería de software, definido por cinco características fundamentales: (11)

- **Dirigido por casos de uso**: El desarrollo está dirigido a satisfacer las necesidades de los usuarios del sistema expresadas en casos de uso.
- **Centrado en la arquitectura**: El desarrollo se centra en una arquitectura bien definida, con relaciones claras entre sus distintos componentes.
- **Iterativo**: El problema y la solución se organizan en pequeñas piezas, de manera que cada iteración se dirige específicamente al desarrollo de un conjunto de ellas.
- **Incremental**: Cada iteración se construye sobre la base creada por las iteraciones anteriores, agregándole capacidades al sistema.
- **Controlado**: El proceso se planifica y en cada momento está claro lo que debe hacerse.

RUP también aumenta la productividad de los desarrolladores mediante acceso a: (11)

- Base de Conocimientos.
- Plantillas.
- Herramientas.

RUP se centra en la producción y mantenimiento de modelos del sistema más que en producir documentos, además de ser una guía para el uso de UML de manera más efectiva.

La metodología RUP está dividida en cuatro fases de desarrollo del producto: (11)

- **Inicio**: El Objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración**: En esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción**: En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial, o sea crear el producto.
- **Transición**: Esta fase se puede subdividir en varias iteraciones, además incluye pruebas del producto para poder hacer el entregable del mismo, así como realizar ajuste menores de acuerdo a ajuste menores propuestos por el usuario. En este punto, la retroalimentación de los usuarios se centra en depurar el producto, configuraciones, instalación y aspectos sobre utilización.

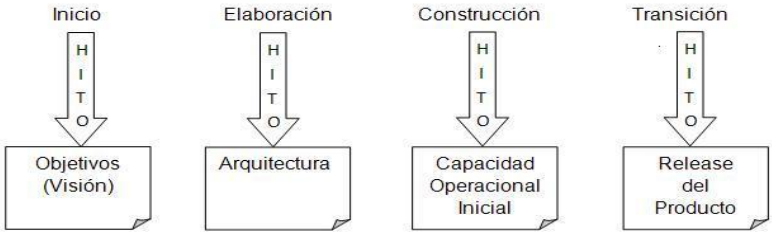



***Figura. 2: Fases e hitos definidos de RUP.***

## *2.1.3. Scrum.*

Scrum divide un proyecto en iteraciones (más conocidos por carreras cortas) de 30 días. Antes de que comience una iteración se define la funcionalidad requerida para esta y entonces se deja al equipo para que la desarrolle y entregue. El objetivo fundamental es estabilizar los requisitos durante las iteraciones. (12)

Sin embargo la gerencia no se desentiende durante la carrera corta. Todos los días el equipo sostiene una reunión corta (quince minutos), llamada scrum, donde el equipo discurre lo que hará al día siguiente. En particular muestran a los bloques de la gerencia: los impedimentos para progresar que se atraviesan y que la gerencia debe resolver. También informan lo que se ha hecho para que la gerencia tenga una actualización diaria del estado del proyecto. (12)

Scrum posee un diseño ideal para adaptarse a los cambios de requisitos. Se considera una metodología eficiente para el desarrollo de proyectos en los que los requisitos no

poseen estabilidad, debido a su capacidad para adaptar el software a las necesidades del cliente en tiempo real.

### *2.1.4. ¿Por qué la metodología de desarrollo de software seleccionada es RUP?*

Lo más importante antes de elegir la metodología que se usará para la construcción de un software, es determinar el alcance que tendrá el proyecto y luego ver cuál es la que más se acomoda. Se ha considerado que Scrum no constituye una metodología factible para el desarrollo del sistema pues se considera una metodología eficiente para aplicar en proyectos con entornos impredecibles, en los que se visualicen cambios constantes y estas características no están presentes en el sistema a desarrollar.

Para determinar la metodología a utilizar se ha decidido establecer una comparación más detallada entre MSF y RUP como se muestra en la siguiente tabla comparativa.

| Atributo | MSF | RUP |
|---|---|---|
| Enfoque iterativo | Si | Si |
| Número de Fases | 5 | 4 |
| Fase inicial | Previendo | Inicio |
| Análisis de fases | Planificación | Elaboración |
| Fase de desarrollo | En desarrollo | Construcción |
| Estabilización de fases | Estabilización | Transición |
| Fase de implementación | Implementación | Transición |
| Flujo de trabajo: Guía de modelos de negocio | No | Si |
| Flujo de trabajo: Guía de requisitos | Si (no detallado) | Si (detallado) |
| Flujo de trabajo: Análisis y guía de diseño | Si (no detallado) | Si (detallado) |
| Flujo de trabajo: Guía de implementación | No | Si |
| Flujo de trabajo: Guía de prueba | No | Si |
| Flujo de trabajo: Guía de implementación | No | Si |
| Flujo de trabajo: Guía de gestión del cambio | No | Si |
| Flujo de trabajo: Proyecto de guía para el manejo | Si (no detallado) | Si (detallado) |
| Flujo de trabajo: Guía para el Medio Ambiente | No | Si |
| Guía de entrevista | No | Si |
| UML guía de uso | No | Si |

***Tabla 1: Comparación de metodologías MSF y RUP.***

Luego de haber analizado las diferentes Metodologías de desarrollo de software comparadas anteriormente se determina utilizar RUP porque sin dudas es más adaptable para diferentes tipos de proyectos, además el equipo de desarrollo está familiarizado con esta metodología, lo que provee una ventaja adicional.

## *2.2. Herramientas CASE[3].*

Se puede definir a las herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante el ciclo de vida de un software. (13)

Las herramientas CASE constituyen aplicaciones informáticas que permiten aumentar la productividad en el desarrollo de software y reducir los gastos en cuanto a tiempo y costo, además de garantizar documentaciones con calidad. Seguidamente se realiza un estudio preliminar de las más usadas para luego seleccionar la más adecuada para utilizar durante el proceso de desarrollo de la herramienta de monitoreo de señales emitidas en la PTARTV.

### *2.2.1. Rational Rose.*

Rational Rose es una herramienta de diseño orientada a objetos, que da soporte al modelado visual, es decir, que permite representar gráficamente el sistema, permitiendo hacer énfasis en los detalles más importantes, centrándose en los casos de uso y enfocándose hacia un software de mayor calidad, empleando un lenguaje estándar común que facilita la comunicación**.** (14)

Rational Rose permite el modelado visual mediante UML de sistemas de software, permite especificar, visualizar y diseñar el sistema antes de codificarlo. Garantiza la consistencia de los modelados de software, permite la generación de documentación y código a partir de los modelos. Además provee capacidades para realizar la ingeniería inversa teniendo el código.

### *2.2.2. Visual Paradigm.*

Visual Paradigm para UML es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software ayuda a obtener una rápida construcción de aplicaciones con calidad y a un menor coste. Visual Paradigm permite modelar los diagramas de clases. Realizar ingeniería inversa si se tiene el código. Además facilita generar el código y la documentación de los diagramas y modelos

[3]Computer Aided Software Engineer. Ingeniería de Software asistida por computadora.

construidos. Esta herramienta permite importar y exportar la documentación en archivos XML.

Es una herramienta colaborativa ya que varios usuarios pueden trabajar sobre el mismo proyecto, lo que facilita el trabajo en equipo. Es una aplicación multiplataforma, diseñado para el trabajo tanto en Windows como en Linux. Posee como peculiaridad sobre el resto de las demás herramientas que cuenta con una potente funcionalidad para la creación de interfaces de usuarios de las aplicaciones.

### *2.2.3. Selección de la herramienta CASE.*

Se decide escoger Visual Paradigm para la construcción del sistema porque es una herramienta CASE que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite la construcción de aplicaciones con calidad y a un menor coste.

Además es una herramienta colaborativa ya que varios usuarios pueden trabajar sobre el mismo proyecto, lo que facilita el trabajo en equipo. Es una aplicación multiplataforma, diseñado para el trabajo tanto en Windows como en Linux. Sumado a los elementos aportados anteriormente, es válido destacar que constituye una de las herramientas líderes en el mundo de la modelación y se ha convertido en la herramienta CASE por excelencia para el desarrollo en entornos libres.

## *2.3. Sistemas Gestores de Bases de Datos (SGBD).*

Se trata de un conjunto de programas no visibles al usuario final que se encargan de la privacidad, la integridad, la seguridad de los datos y la interacción con el sistema operativo. Proporciona una interfaz entre los datos, los programas que los manejan y los usuarios finales. Cualquier operación que el usuario hace contra la base de datos está controlada por el gestor.

Los sistemas gestores de base de datos se pueden definir como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas

de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad. (15)

Es importante conocer las características además de las ventajas y desventajas de cada sistema gestor de base de datos. Se debe analizar los productos que más sobresalen de acuerdo a las necesidades, optando por la mejor opción. Algunas de estas alternativas son Oracle, Microsoft SQL Server y MySQL, que comercialmente son más fuertes y en el mundo del software libre, teniendo opciones tan completas como SQL Server 2000 y PostgreSQL que son gestores muy usados.

### *2.3.1. PostgreSQL.*

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

PostgreSQL posee técnicas para programar del lado de servidor y el control multiusuario. Las interfaces de PostgreSQL a C, C ++, ODBC, JDBC y Perl, incluyen reservas, solución y acceso a la configuración. Está ampliamente considerado como uno de los SGBD de código abierto más avanzado del mundo. (16)

### *2.3.2. MySQL.*

MySQL cuenta con millones de aplicaciones y aparece en el mundo informático como una de las más utilizadas por usuarios del medio. El programa MySQL se usa como servidor a través del cual pueden conectarse múltiples usuarios y utilizarlo al mismo tiempo. En las últimas versiones se pueden destacar las siguientes características principales: (17)

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: uno de estructura, uno de datos y uno de índice que soporta hasta 32 índices por tabla.
- Flexible sistema de contraseñas y gestión de usuarios, con un muy buen nivel de seguridad en los datos.

- El servidor soporta mensajes de error en distintos idiomas.

### *2.3.3. ¿Por qué PostgreSQL?*

Se propone el uso de PostgreSQL como sistema gestor de base de datos ya que es considerado como uno de los gestores de base de datos de código abierto más avanzado del mundo, se caracteriza por su robustez y escalabilidad, además facilita diferentes características que se encontraban solamente en gestores de base de datos comerciales de difícil acceso, como por ejemplo Oracle que tiene como mayor inconveniente que es un software privativo y las licencias son extremadamente caras.

PostgreSQL es un servidor de base de datos relacional libre, liberado bajo la licencia BSD. Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2. Mediante un sistema denominado MVCC (Acceso concurrente multiversión) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta. Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido. Además provee soporte para: números de precisión arbitraria, texto de largo ilimitado, figuras geométricas (con una variedad de funciones asociadas), direcciones IP, claves ajenas también denominadas llaves ajenas o llaves foráneas (foreign keys), vistas, herencia de tablas, tipos de datos y operaciones geométricas.

## *2.4. Lenguajes de Programación.*

Para poder interpretar las instrucciones que se le dan a una computadora es necesario un lenguaje propio que permita controlar el comportamiento de la comunicación del hombre con la máquina en general, para ello se utilizan los lenguajes de programación. Existen diferentes lenguajes de programación, cada uno posee sus características específicas. Entre los más sobresalientes se conocen algunos como C++, Java, C#, ASP, PHP.

### *2.4.1. C++.*

Desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido ya que provee la programación estructurada y la programación orientada a objetos. Por esto se suele decir que el C++ es un lenguaje de programación multiparadigma. (18)

El nacimiento de C++ se sitúa en el año 1980, cuando Bjarne Stroustrup, de los laboratorios Bell, desarrolló una extensión de C llamada “C with Classes" que permitía aplicar los conceptos de la programación orientada a objetos con el lenguaje C. Tomó ideas de otros lenguajes importantes de la época como ALGOL68 o ADA. Durante los siguientes años, Stroustrup continuó el desarrollo del nuevo lenguaje y en 1983 se acuñó el término C++. (18)

Las principales ventajas que presenta el lenguaje C++ son: (18)

- **Difusión:** Al ser uno de los lenguajes más empleados en la actualidad, posee un gran número de usuarios y existe una gran cantidad de libros, cursos y páginas web dedicadas a él.
- **Versatilidad:** C++ 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:** C++ es uno de los lenguajes más rápidos en cuanto ejecución.
- **Herramientas:** Existe una gran cantidad de compiladores, depuradores, librerías, entre otros.

Este lenguaje cuenta con la ventaja, a excepción del ensamblador, de generar los programas más compactos y rápido. El código es portable y podrá ejecutarse en cualquier máquina y bajo cualquier sistema operativo. Y si es necesario, proporcionan un acceso a bajo nivel de hardware sólo igualado por el ensamblador. Por estas características y las anteriormente señaladas se selecciona para la implementación de la solución propuesta el lenguaje de programación C++.

### *2.4.2. C#.*

Combina los mejores elementos de múltiples lenguajes de amplia difusión como C++, Java, Visual Basic o Delphi y permite al igual que estos lenguajes programar fácilmente a bajo nivel. Es un lenguaje orientado a objetos, posee una sintaxis similar

a la de C++, implementa una fuerte política de seguridad de tipos de datos, elimina la utilización de punteros aunque mantiene una reserva en caso extremo de su uso.

Posee mecanismos como los índices y la instrucción foreach, que hacen más fácil e intuitivo el trabajo, elimina la herencia múltiple (ofrece el uso de interfaces) y facilita el trabajo con propiedades y eventos.

A continuación, algunas características de este lenguaje que lo hacen generar componentes de sistema duraderos:

- Gran robustez, gracias a la recolección de elementos no utilizados y a la seguridad en el tratamiento de tipos.
- Seguridad implementada por medio de mecanismos de confianza intrínsecos del código.
- Plena compatibilidad con conceptos de meta datos extensibles.
- Interactúa con otros lenguajes, entre plataformas distintas, y con datos heredados debido a la plena interoperabilidad por medio de los servicios de COM+ 1.0 y .NET Framework con un acceso limitado basado en bibliotecas, compatibilidad con XML para interacción con componentes basados en tecnología Web y capacidad de control de versiones para facilitar la administración y la implementación. (19)

### *2.4.3. Selección del lenguaje de programación.*

En la actualidad, el C++ es un lenguaje versátil, potente y general. Su éxito entre los programadores profesionales le ha llevado a ocupar el primer puesto como herramienta de desarrollo de aplicaciones. El C++ mantiene ventajas en cuanto a riqueza de operadores y expresiones, flexibilidad, concisión y eficiencia. La evolución de C++ ha continuado con la aparición de Java, un lenguaje creado simplificando algunas cosas de C++ y añadiendo otras, que se utiliza para realizar aplicaciones en Internet.

El C++ es a la vez un lenguaje orientado a algoritmos y orientado a objetos. Las características propias de la programación orientada a objetos POO de C++ son modificaciones mayores que cambian radicalmente su naturaleza. (20)

Debido a las características y tomando las principales ventajas que posee este lenguaje que lo hacen poderoso ante los demás, es que se decide seleccionar el mismo para realizar la implementación de la aplicación.

## *2.5. Entorno de Desarrollo Integrado (IDE).*

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, poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, 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. Los IDEs proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Java, C#. Algunos ejemplos de entornos integrados de desarrollo (IDE) son los siguientes: KDevelop, QtCreator, Eclipse, NetBeans. (21)

### *2.5.1. QtCreator.*

Qt es un framework de desarrollo de aplicaciones multiplataforma. Viene acompañado de un conjunto de herramientas para facilitar su uso. El mismo incluye una colección de contenedores genéricos. Lo contenedores de Qt disponen de dos tipos de iteradores: estilo STL, y estilo Java, que apuntan entre los elementos y no a los elementos. Qt incluye una serie de funcionalidades que no están normalmente en C++. Para ello usa su propio preprocesador llamado MOC (Meta-Object Compiler). Qt usa en gran parte de su API un tipo especial de objetos no nativo de C++, los meta objetos, que se caracterizan por signals y slots. (22)

Una de las principales ventajas de QtCreator es que permite que un equipo de desarrolladores pueda compartir proyectos a través de diferentes plataformas de desarrollo (Microsoft Windows y Linux) con una herramienta común para el desarrollo y depuración. Qt busca la simplicidad, facilidad de uso, la productividad, extensibilidad. Las características principales de QtCreator permiten a los desarrolladores realizar las tareas siguientes: (22)

- Comenzar con el desarrollo de aplicaciones Qt rápida y fácilmente con asistentes de proyectos, y acceder rápidamente a los últimos proyectos y sesiones.
- Usando módulo QtPhonon proporciona un entorno de trabajo multiplataforma multimedia que permite el uso de audio y vídeo en aplicaciones Qt.
- Se le puede incluir la librería libvlc, la cual permite el trabajo con videos y audio, además de ofrecer una amplia variedad de librerías incluidas con funcionalidades ya implementadas que facilitan el trabajo.
- Diseño de aplicaciones basado en interfaz de usuario con el editor integrado, QtDesigner.
- Desarrollo de aplicaciones con C ++ por su avanzado editor de código que proporciona características de gran alcance para completar fragmentos de código, el código de refactorización, y ver el contorno de los archivos (es decir, la jerarquía de símbolo de un archivo).
- Generar, ejecutar e implementar proyectos de Qt que se dirigen de escritorio y plataformas móviles, como Microsoft Windows, Mac OS X, Linux, Symbian y Maemo.
- Depurar con la GNU y el BDC mediante una interfaz gráfica de usuario con mayor conocimiento de la estructura de clases de Qt.
- Fácil acceso a la información con el sistema de ayuda contextual integrada Qt.

### *2.5.2. NetBeans.*

El IDE NetBeans es un entorno integrado de desarrollo galardonado disponible para Windows, Mac, Linux y Solaris. NetBeans consiste en un IDE de código abierto y una plataforma de aplicaciones que permiten a los desarrolladores crear rápidamente web, para empresas, desarrollar aplicaciones de escritorio y aplicaciones móviles utilizando la plataforma Java, así como JavaFX, PHP, JavaScript y Ajax, y C / C + +. Además es apoyada por una comunidad de desarrolladores y ofrece una amplia documentación y recursos de capacitación, así como una variada selección de plugins de terceros. (23)

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de Java escritas para interactuar con las API de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones

construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software. (24)

### *2.5.3. ¿Por qué QtCreator?*

Para el desarrollo de la aplicación se selecciona el IDE QtCreator ya que posee varias ventajas sobre los demás IDEs antes mencionado, pues QtCreator ofrece un entorno completo de desarrollo para la creación de aplicaciones. Es una herramienta ligera con un enfoque estricto de las necesidades de los desarrolladores, garantiza la productividad y usabilidad de dichas aplicaciones.

QtCreator es multiplataforma, permite la colaboración entre los diseñadores y desarrolladores. Los diseñadores trabajan en un entorno visual, mientras que los desarrolladores trabajan en un IDE con todas las funciones ya que QtCreator es compatible desde el diseño hasta el código. Unas de las cosas novedosas que pose QT es la versión incorporada de QtDesigner. Su inclusión significa que usted puede arrastrar los widgetsQt en el lienzo en blanco de la aplicación y cambiar el código fuente de inmediato.

## *2.6. Conclusiones.*

En el capítulo se ha realizado un estudio de las tecnologías necesarias para el desarrollo de la herramienta para el monitoreo de señales emitidas en la PTARTV. Para esto se han caracterizado cada una de ellas y se tomó en consideración el llamado de la dirección del país y de la UCI a la soberanía tecnológica para seleccionar tecnologías libres. Se puede concluir partiendo, de que no existe una tecnología superior, sino que cada una tiene sus propias características, que la convierte en la opción factible de utilizar según las peculiaridades del proyecto que se desee desarrollar. Teniendo en cuenta esto se ha determinado que las tecnologías más factibles para el desarrollo del proyecto son las que se relacionan en la siguiente tabla:

| | |
|---|---|
| Metodología de desarrollo | RUP |
| Herramienta CASE | Visual Paradigm |
| Sistema Gestor de Base de Datos | Postgres SQL |
| Lenguaje de programación | C++ |
| Entorno de desarrollo integrado | QtCreator |

***Tabla 2: Tecnologías factibles para el desarrollo.***

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

## 3. *Introducción.*

En el presente capítulo se realiza la descripción de la solución propuesta. La misma se describe a través de un modelo de negocio, puesto que el sistema tiene bien definido los procesos del negocio. Además se enumeran los requisitos Funcionales y No Funcionales que debe tener el sistema que se propone. Luego se realiza una concepción general del sistema, permitiendo identificar mediante un Diagrama de Casos de Uso, las relaciones de los actores que interactúan con el sistema, y las secuencias de acciones que se ejecutan, realizando la descripción de los casos de uso del sistema. También se realizan los diagramas de análisis que permitirán un mejor entendimiento de las funcionalidades del sistema para el desarrollo del diseño del mismo.

## *3.1. Entorno donde trabajará el sistema.*

La herramienta de monitoreo se empleará en la Dirección de Televisión Universitaria (DTU). Precisamente esta dirección es la encargada de transmitir el flujo de información a través de PTARTV, para los medios televisivos y radiales, demostrando la necesidad de una aplicación informática para el monitoreo de las señales de televisión que se trasmiten.

## *3.2. Modelo de Negocio.*

Al aplicar la metodología seleccionada para el desarrollo de software, que incluye realizar análisis y diseño, se hace necesario realizar el modelado del negocio, pues el mismo permite modelar las operaciones de la organización. Brindando la posibilidad de comprender los procesos que verdaderamente existen, así como quién lo dirige y quiénes están involucrados en cada proceso. (25)

El modelo del negocio es un proceso complejo que depende de la comunicación entre clientes, especialistas y de la relación dentro de los grupos de trabajo, entre otros factores. Un sistema que no responda a las necesidades de los clientes no cumple los requisitos mínimos de calidad. Por todo lo planteado anteriormente se hace indispensable usar herramientas que permitan a los analistas obtener un Modelo del

Negocio que se ajuste a las necesidades de los clientes y que utilice la experiencia acumulada hasta el momento en sistemas con características similares. (25)

Realizar una buena modelación del negocio garantiza que los requisitos predeterminados por el cliente sean bien comprendidos por los desarrolladores y así la satisfacción del cliente será la deseada. Una buena comprensión del negocio requiere determinar qué hace y qué no hace la empresa además de cómo crear una propuesta atractiva y de valor para los clientes.

### *3.2.1. Actores del Negocio.*

Cuando se realiza el modelado del negocio primero se debe determinar los actores del mismo. Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos, con los que el negocio interactúa. Un actor no representa un usuario físico sino el rol específico que interactúa con el sistema y recibe algún beneficio del mismo. Este siempre va a permanecer fuera de las fronteras del negocio que se investiga. (26)

| Actor | Justificación |
|---|---|
| Jefe de Área de Trasmisiones. | Directivo de la DTU que se desempeña al frente del área de trasmisión de canales. |

***Tabla 3: Actores del negocio.***

### *3.2.2. Trabajadores del negocio.*

Un trabajador del negocio puede ser un sistema, persona o software que representa un rol y realiza las actividades comprendidas en los procesos del negocio. Puede colaborar con otros trabajadores del negocio y manipula las entidades del negocio para realizar sus responsabilidades. Son identificados dentro de las fronteras del negocio y en un futuro se convertirán en usuarios del sistema que se quiere construir.

| Trabajador | Justificación |
|---|---|
| Transmisor | Es la persona encargada de monitorear manualmente los canales. |

***Tabla 4: Trabajadores del negocio.***

### *3.2.3. Diagramas de Casos de Usos del Negocio.*

La forma de representar y almacenar las experiencias obtenidas a través del modelado del negocio se realizan a través de casos de uso. Un caso de uso mantiene todos los atributos y características relevantes de un evento. Estas características servirán como índices para la recuperación de información a lo largo del ciclo de vida del proceso.

Una vez definidos los actores, los trabajadores y los procesos de negocio es posible construir el Diagrama de Casos de Uso del Negocio. El mismo describe los procesos de negocio de una empresa en términos de casos de uso del negocio y actores del negocio que se corresponden con los procesos del negocio y los clientes, respectivamente (Figura. 3).


***Figura. 3: Diagrama de Casos de Usos del Negocio***

### *3.2.4. Descripción de los Casos de Uso del Negocio.*

| Nombre del Caso de Uso | Monitorear Canal |
|---|---|
| Actores | Jefe de Área de Transmisión |
| Propósito | Permite monitorear los canales de televisión |
| Resumen | El caso de uso se inicia cuando el jefe del área de transmisión necesita información del estado de los canales de televisión que están al aire y solicita el transmisor que es el encargado de revisar si los canales se están transmitiendo o no, el mismo una vez recibida la solicitud comienza a revisar canal por canal para así obtener información de las señales, en caso de detectar errores envía un informe plasmando los problemas que encontró, de lo contrario no se envía información |

| | |
|---|---|
| | referente. |
| Casos de Usos Asociados | - |
| Curso Normal de Eventos | |
| Acción del Actor | Respuesta del Negocio |
| 1. El Jefe del Área de transmisión solicita monitorear canales de televisión | 2. El transmisor recibe la solicitud de monitorear los canales.<br>3. El transmisor enciende el televisor y comienza a explorar canal por canal para ver si los mismos se están trasmitiendo y si las condiciones son favorables (audio y video).<br>4. Detecta errores en la transmisión<br>5. Envía reporte de estado de la transmisión.<br>6. Ejecuta el paso 3 cada un tiempo determinado. |
| 7. Recibe reporte | |
| Curso Alterno | |
| Acción del Actor | Respuesta del Negocio |
| | 4. No detecta errores en la Transmisión.<br>5. No envía reporte. |

***Tabla 5: Descripción del Caso de Uso del Negocio***

### *3.3.5. Diagrama de Actividades del Caso de Uso del Negocio Monitorear Canal.*

El diagrama de actividades representa el comportamiento interno de una operación o de un caso de uso, bajo la forma de un desarrollo por etapas, agrupadas secuencialmente. Tiene como propósito fundamental: modelar el flujo de tareas y modelar las operaciones. (27)

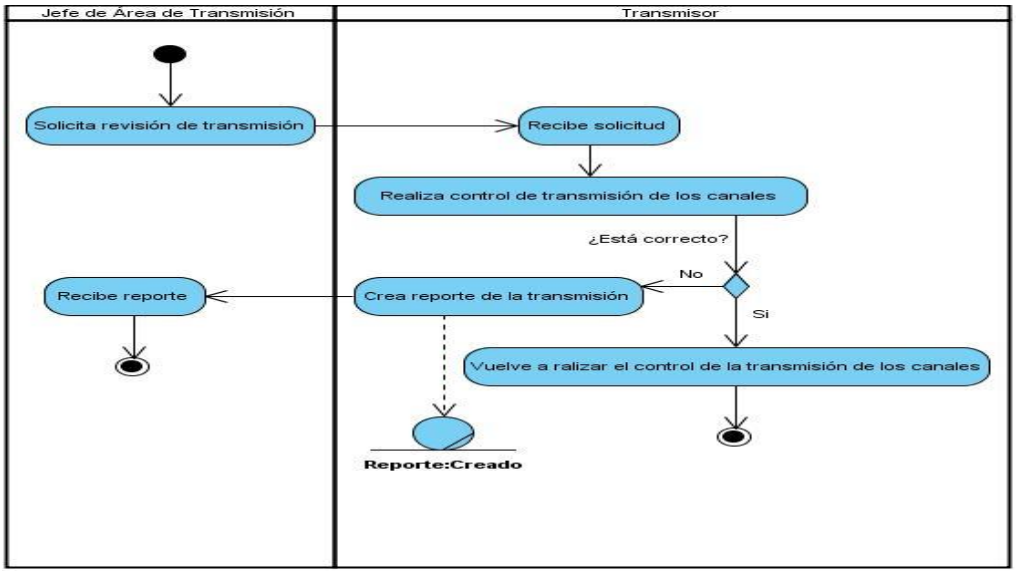



***Figura. 4: Diagrama de Actividad CUN Monitorear Canal.***

### *3.3.6. Modelo de Objetos del Negocio.*

Este modelo permite representar la realidad a imagen y semejanza, considerando que el mundo real que tratamos está constituido por cosas, entidades u objetos que interactúan entre sí (Figura 5).

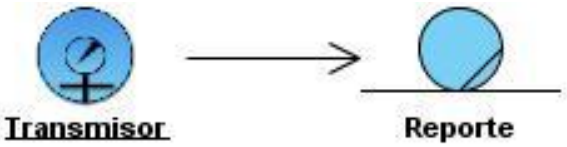



***Figura. 5: Modelo de Objetos.***

## *3.4. Requisitos.*

Cuando se comienza el desarrollo de una aplicación de software, es de suma importancia realizar encuentros con el cliente para poder definir cuáles son los objetivos y funcionalidades que debe cumplir el mismo. La fase de análisis consiste en detectar estas funcionalidades y expresarlas en forma de requisitos. El análisis de requisitos es la primera etapa de un proyecto software, en ella se tratan de definir las condiciones o capacidades necesarias para uno o varios usuarios con el fin de solucionar un problema o conseguir un objetivo.

Luego de haber enfatizado en los conceptos importantes que rodean al objeto de estudio, se puede comenzar a analizar ¿Qué condiciones o funciones debe cumplir el sistema para que se cumplan los objetivos planteados? Para ello existen dos tipos de requisitos:

- ***Funcionales*** (las funciones que el sistema deberá ser capaz de realizar. Dentro de ellos se incluyen las acciones que podrán ser ejecutadas por el usuario, las acciones ocultas que debe realizar el sistema, y las condiciones extremas a determinar por el sistema).
- ***No Funcionales*** (las condiciones extremas a determinar por el sistema, además las características que debe tener la aplicación).

### *3.4.1. Requisitos Funcionales.*

**R.1** Autenticar Usuario.
**R.2** Monitorear Canal.
**R.3** Visualizar Programación del Día.
**R.4** Visualizar Media Seleccionada.
**R.5** Visualizar Canal a Pantalla Completa.
**R.6** Visualizar Datos del Canal.
**R.7** Actualizar Transmisión.
**R.8** Notificar Error en la Transmisión.
**R.9** Seleccionar Configuración de Entorno de Trabajo.
**R.10** Visualizar Reporte de Errores.

### *3.4.2. Requisitos No Funcionales.*

#### 3.4.2.1. *Usabilidad.*

- El sistema debe exigir a los usuarios autenticarse antes de poder usar sus funcionalidades.
- El sistema podrá ser utilizado solo por las personas autorizadas por el administrador.
- Facilidad de uso.
- Además garantizará una conexión rápida y segura con la base de datos que contendrá la información, lo que permitirá facilidades de actualización.

#### *3.4.2.2. Rendimiento.*

- El sistema debe ser capaz de brindar la información necesaria y mostrar la información según lo seleccionado.

- La eficiencia del producto estará determinada por al velocidad de transmisión de la red además de su nivel de congestión.

### *3.4.2.3. Seguridad.*

- El ingreso a la aplicación será mediante el mecanismo de autenticación de usuarios, permitiendo acceder solamente al personal autorizado a utilizar el mismo.
- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos, las personas que no están registradas no podrán acceder al sistema.

### *3.4.2.4. Utilización de recursos.*

- El sistema utilizará BD para obtener las medias, los listados de la programación, los datos de los canales y guardar los reportes de errores.
- Capacidad de respuesta, obtener los resultados a mostrar en el menor tiempo posible.

### *3.4.2.5. Disponibilidad.*

- El sistema estará disponible las 24 horas del día permitiendo su uso a todo momento.

### *3.4.2.6. Soporte.*

- El sistema permitirá su extensibilidad, pudiéndosele agregar nuevos funcionalidades o modificar los existentes para lograr mejores prestaciones en el momento que se requieran realizar cambios.
- Las pruebas del sistema se realizarán en la Dirección de Televisión Universitaria (DTU), las mismas permitirán evaluar en la práctica las funcionalidades y las ventajas de este nuevo producto, así como los errores.
- El departamento GEYSED que incluye a su vez el proyecto PTARTV será en encargado del mantenimiento de la aplicación. Además garantizará la integridad de los datos que se almacenan en la base de datos.
- La información deberá estar disponible a los usuarios en todo momento, limitada solamente por las restricciones.

### *3.4.2.7. Software.*

- Sistema Operativo Linux Ubuntu 10.10 u otras versiones superiores a esta.
- Gestor de Base Datos Postgres SQL.

### *3.4.2.8. Hardware.*

- Core 2 Duo 2.20 GHz como mínimo.
- HDD 40 GB como mínimo.
- Tarjeta de Red Gigabit Ethernet 1 Gbps o superior.
- 2 GB de memoria RAM como mínimo.

### *3.4.2.9. Interfaz o Apariencia.*

- El sistema exhibirá una apariencia profesional, sin gran cantidad de imágenes, debe ser ágil, intuitiva, muy legible y simple de usar, teniendo en cuenta las características de los usuarios hacia los cuales va dirigido el mismo.
- La interfaz estará diseñada de modo tal que el usuario pueda tener en todo momento el control de la aplicación, lo que le permitirá ir de un punto a otro dentro de ella con facilidad.
- Se garantizará que la aplicación sea lo más interactiva posible. Se usarán los colores: blanco, negro y gris en diferentes tonalidades.

## *3.5. Descripción del sistema propuesto.*

Una vez realizado el modelo de negocio y analizado los requisitos funcionales y no funcionales del software, se hace posible modelar y proponer el sistema que se desea desarrollar.

### *3.5.1. Descripción de los actores del sistema.*

Un actor del sistema es aquel que representa un rol, el cual es asumido por cualquier sistema o persona que interactúe con el sistema que se desea desarrollar. En la siguiente tabla se especifican los actores y se le da una breve descripción de los mismos.

| Actor | Descripción |
|---|---|
| Transmisor | Encargado de monitorear las señales que se están transmitiendo. |

| | |
|---|---|
| Sistema | Es un actor ficticio el cual se representa para los procesos automáticos del sistema. |

***Tabla 6: Actores del sistema.***

### *3.5.2. Diagrama de Casos de Uso del Sistema.*

En la modelación del sistema fueron agrupados los requisitos funcionales, permitiendo conformar los casos de usos del mismo. Estos representan los artefactos narrativos que describen el comportamiento existente entre los actores y las funcionalidades del sistema.

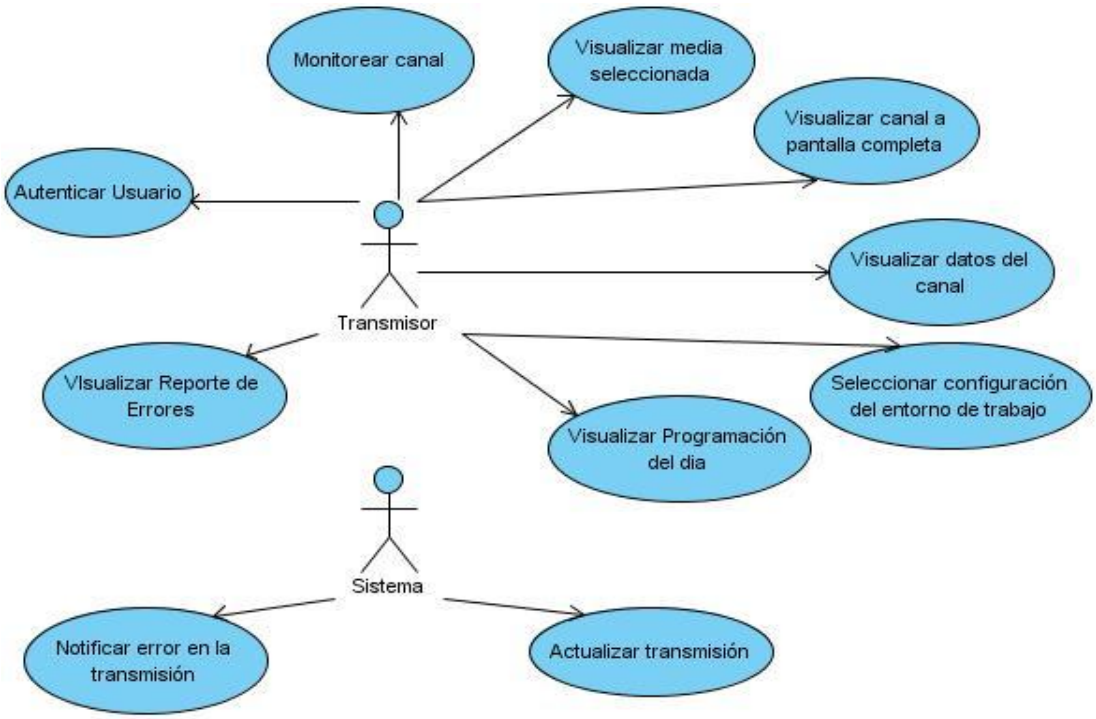



***Figura. 6: Diagrama de Caso de Uso del Sistema***

### *3.5.3. Especificación de los Casos de Uso.*

A continuación se realiza la descripción correspondiente a cada uno de los casos de uso del sistema. Además de una detallada especificación de las acciones del mismo, lo cual explicará una visión general del comportamiento que ostentará el sistema a implementar.

### *3.5.3.1. Descripción del Caso de Uso Autenticar Usuario.*

| | |
|---|---|
| **Caso de Uso:** | Autenticar usuario |
| **Actores:** | Usuario |
| **Resumen:** | El caso de uso se inicia cuando el usuario introduce los datos que se solicitan para acceder a la aplicación, estos se verifican y el mismo finaliza dándole los permisos, en caso de poseer los privilegios necesarios. |
| **Precondiciones:** | El usuario debe estar registrado en la Base de Datos. |
| **Referencias** | R.1 |
| **Prioridad** | Crítico |
| **Flujo Normal de Eventos** | |
| **Acción del Actor** | **Respuesta del Sistema** |
| 1. Introduce su usuario y contraseña en los campos de autenticación, (A) y (B) respectivamente.<br>2. Oprime el botón (C), para solicitar su entrada al sistema. | 3. Verifica las credenciales del usuario (usuario, contraseña, rol).<br>4. Finaliza el caso de uso cuando se muestra la interfaz correspondiente a la herramienta (F). |
| **Flujos Alternos "2"** | |
| **Acción del Actor** | **Respuesta del Sistema** |
| 2. Oprime el botón (D), para cancelar su solicitud de entrada al sistema. | 3. Termina el caso de uso cuando el sistema cancela su autenticación. |
| **Flujos Alternos "3"** | |
| **Acción del Actor** | **Respuesta del Sistema** |
| 4. El transmisor oprime el botón (E), cancelando el mensaje de error. | 3. El sistema muestra un mensaje de error debido a que la autenticación no es de un usuario autorizado.<br>5. El sistema cierra el mensaje y muestra nuevamente la interfaz de autenticación (Z). |

**Prototipo de interfaz**

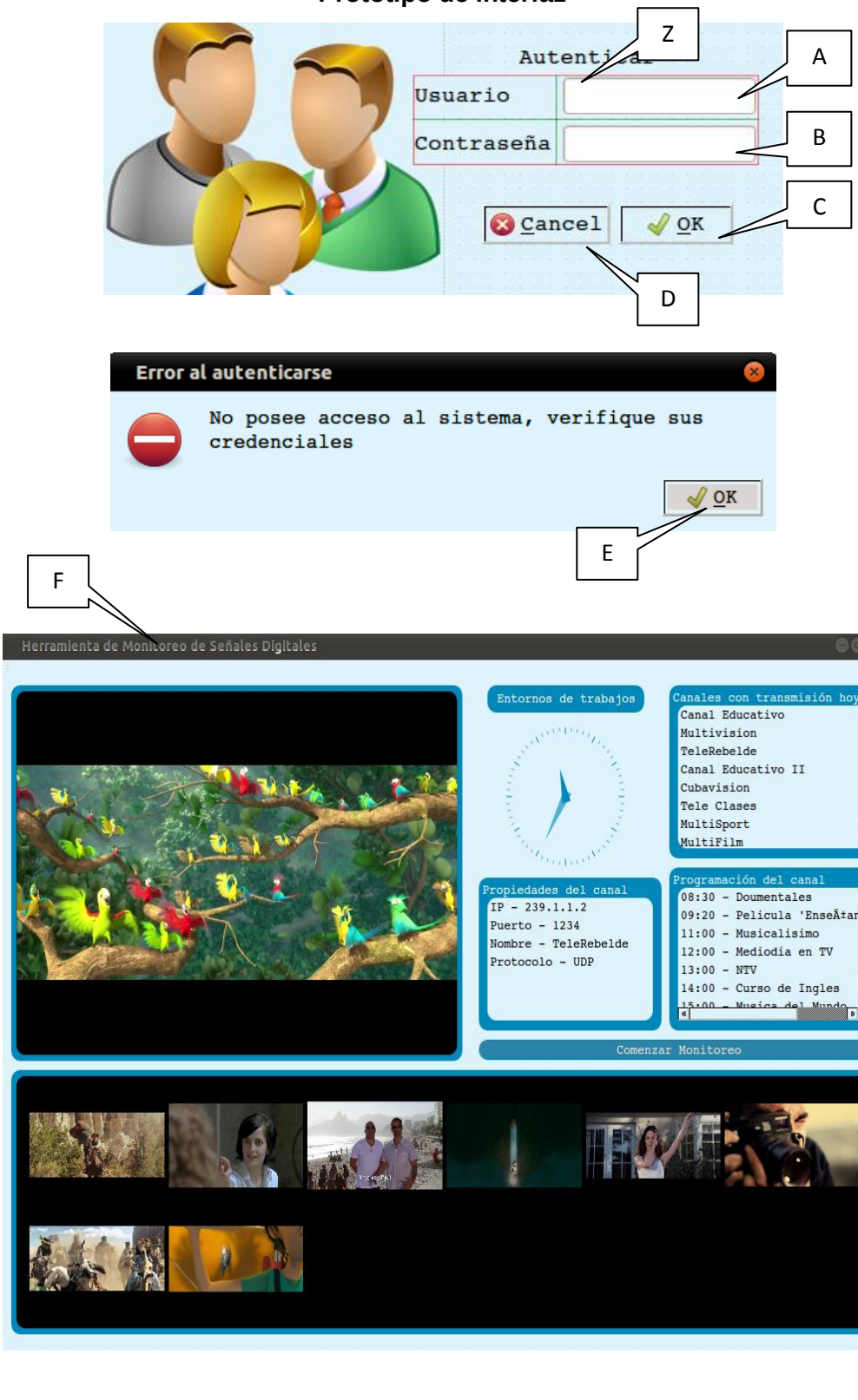



*Tabla 7: Descripción del Caso de Uso Autenticar Usuario*

### *3.5.3.2. Descripción del Caso de Uso Monitorear Canal.*

| | |
|---|---|
| **Caso de Uso:** | Monitorear Canal. |
| **Actores:** | Transmisor. |
| **Resumen:** | El caso de uso comienza cuando el transmisor oprime el botón comenzar monitoreo, monitoreando la emisión de uno o varios canales. El caso de uso finaliza cuando dejan de ser transmitidos. |
| **Precondiciones:** | Tiene que existir conexión con la base de datos.<br>El transmisor tiene que estar autenticado. |
| **Referencias** | R.2 |
| **Prioridad** | Crítico |

| **Flujo Normal de Eventos** | |
|---|---|
| **Acción del Actor** | **Respuesta del Sistema** |
| 1. El caso de uso comienza cuando el transmisor oprime el botón comenzar el monitoreo (G).<br><br>3. Finaliza el caso de uso cuando deja de existir algún canal en trasmisión. | 2. El sistema muestra en su interfaz:<br>a) Lista de los canales que están en transmisión (A).<br>b) La planificación de los canales que se están transmitiendo (B).<br>c) Los datos de los canales que están en trasmisión (C).<br>d) La hora del sistema (D).<br>e) Video de los canales que están en transmisión (E).<br>f) Reproducción de una de las señales con mayor resolución (F). |


*Tabla 8: Descripción del Caso de Uso Monitorear Canal*

### *3.5.3.3. Descripción del Caso de Uso Visualizar Programación del Día.*

| | |
|---|---|
| **Caso de Uso:** | Visualizar Programación del día |
| **Actores:** | Transmisor |
| **Resumen:** | El caso de uso se inicia en cuanto se selecciona uno de los canales que poseen transmisión, se ejecuta mostrando la planificación que tienen los canales, finaliza el caso de uso cuando se muestran las transmisiones que tendrá el canal seleccionado. |
| **Precondiciones:** | El transmisor tiene que estar autenticado.<br>Debe existir conexión con la base de datos<br>Debe existir algún canal en transmisión.<br>El transmisor tiene que seleccionar uno de los canales que se están transmitiendo. |
| **Referencias** | R.3 |
| **Prioridad** | Critico |
| **Flujo Normal de Eventos** | |

| Acción del Actor | Respuesta del Sistema |
|---|---|
| 1. El caso de uso se inicia cuando se selecciona uno de los canales que tienen transmisión para el día (A). | 2. El sistema muestra la programación del día planificada para el canal que se ha seleccionado, mostrando su hora de comienzo además del nombre del mismo (B).<br>3. El caso de uso finaliza cuando se muestra toda la programación que tiene en canal que fue seleccionado. |
| **Prototipo de Interfaz**<br>  | |

*Tabla 9: Descripción del Caso de Uso Visualizar Programación del Día*

## *3.5.3.4. Descripción del Caso de Uso Visualizar Media Seleccionada.*

| | |
|---|---|
| **Caso de Uso:** | Visualizar Media Seleccionada |
| **Actores:** | Transmisor |
| **Resumen:** | El caso de uso se inicia cuando el transmisor del sistema selecciona unos de los canales que se están visualizando, este se visualiza a mayor resolución que su estado inicial finalizando cuando se muestra en el extremo superior izquierdo de la aplicación. |
| **Precondiciones:** | El trasmisor tiene que estar autenticado.<br>El canal debe estar en transmisión. |
| **Referencias** | R.4 |
| **Prioridad** | Critico |

| Flujo Normal de Eventos | |
|---|---|
| **Acción del Actor** | **Respuesta del Sistema** |
| 1. El caso de uso se inicia cuando el transmisor da clic sobre una de las | 2. El sistema visualiza la media seleccionada a mayor resolución |

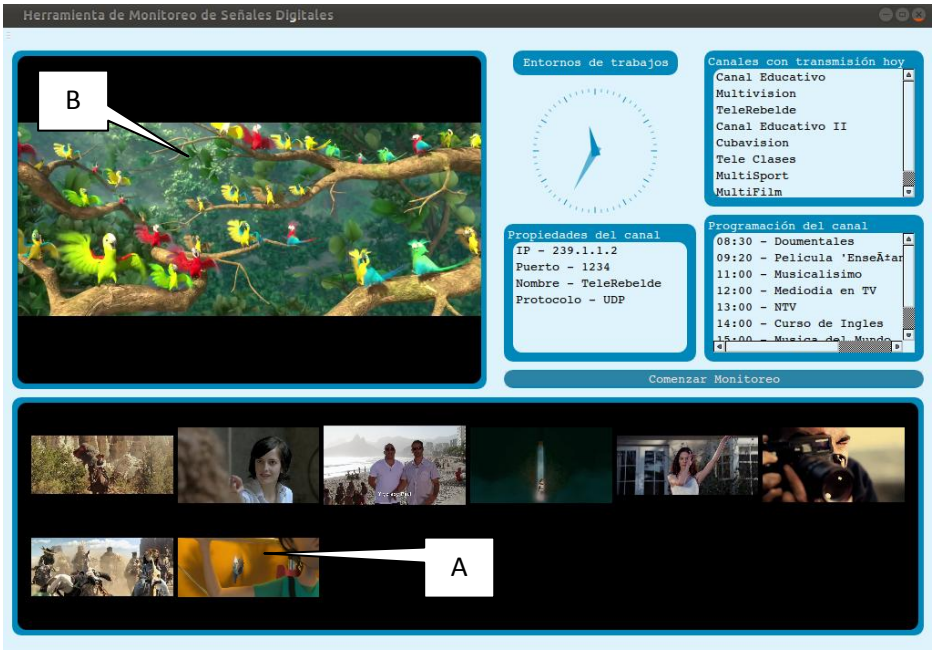
| | |
|---|---|
| medias que se están transmitiendo (A). | (B).<br>3. El caso de uso finaliza cuando se muestra la media con mayor resolución. |
| **Prototipo de Interfaz**<br>  | |

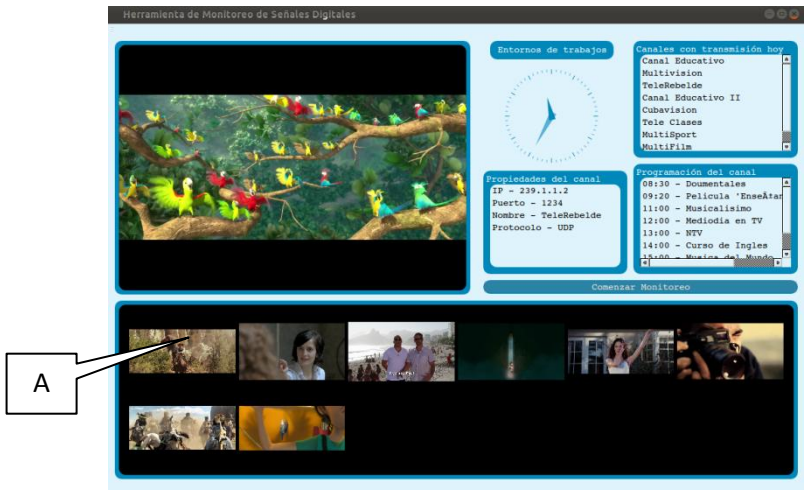
*Tabla 10: Descripción del Caso de Uso Visualizar Media Seleccionada*

### *3.5.3.5. Descripción del Caso de Uso Visualizar Canal a Pantalla Completa.*

| | |
|---|---|
| **Caso de Uso:** | Visualizar canal en pantalla completa |
| **Actores:** | Transmisor |
| **Resumen:** | El caso de uso comienza cuando el transmisor da doble clic sobre uno de los canales, visualizándose en pantalla completa, finalizando cuando da nuevamente doble clic para mostrar la interfaz correspondiente al monitoreo. |
| **Precondiciones:** | El transmisor tiene que estar autenticado.<br>Tiene que existir media en transmisión. |
| **Referencias** | R.5 |
| **Prioridad** | Critico |
| **Flujo Normal de Eventos** | |
| **Acción del Actor** | **Respuesta del Sistema** |
| 1. El caso de uso inicia cuando el | 2. El sistema muestra el canal |

| | |
|---|---|
| transmisor da doble clic sobre una de las medias que se están transmitiendo (A).<br>3. El caso de uso finaliza cuando el transmisor le da doble clic a la media que esta visualizándose en pantalla completa y nuevamente se visualiza la interfaz correspondiente a la herramienta de monitoreo. | que el transmisor seleccionó a pantalla completa (B). |

**Prototipo de Interfaz**



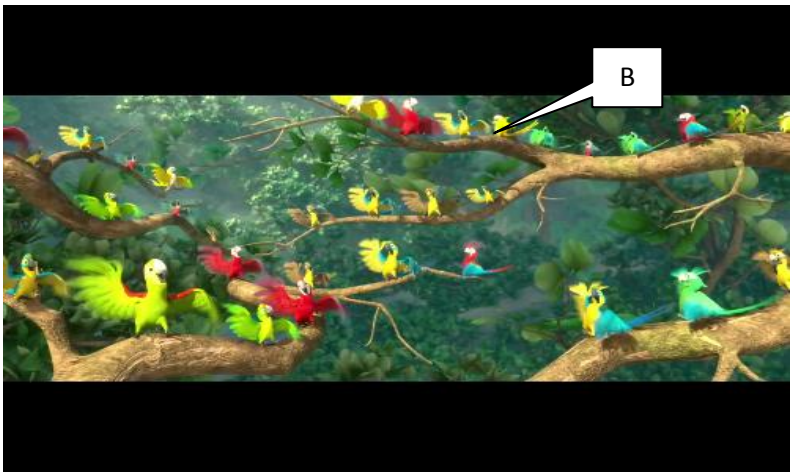




*Tabla 11: Descripción del Caso de Uso Visualizar Canal a Pantalla Completa*

## *3.5.3.6. Descripción del Caso de Uso Visualizar Datos del Canal.*

| | |
|---|---|
| **Caso de Uso:** | Visualizar Datos del Canal |
| **Actores:** | Transmisor |

| | |
|---|---|
| **Resumen:** | El caso de uso se inicializa cuando el transmisor selecciona uno de los canales que están en transmisión para el día, al seleccionarlo se muestran todos los datos del mismo. Finalizando cuando se muestran los datos de los canales. |
| **Precondiciones:** | El transmisor tiene que estar autenticado.<br>Tiene que existir programación para el día. |
| **Referencias** | R.6 |
| **Prioridad** | Critico |

| **Flujo Normal de Eventos** | |
|---|---|
| **Acción del Actor** | **Respuesta del Sistema** |
| 4. El caso de uso se inicializa cuando se selecciona uno de los canales que están en transmisión (A). | 5. El sistema muestra los datos del canal seleccionado (B).<br>6. Finaliza el caso de uso cuando se muestran todos los datos del canal que se seleccionó. |

**Prototipo de Interfaz**

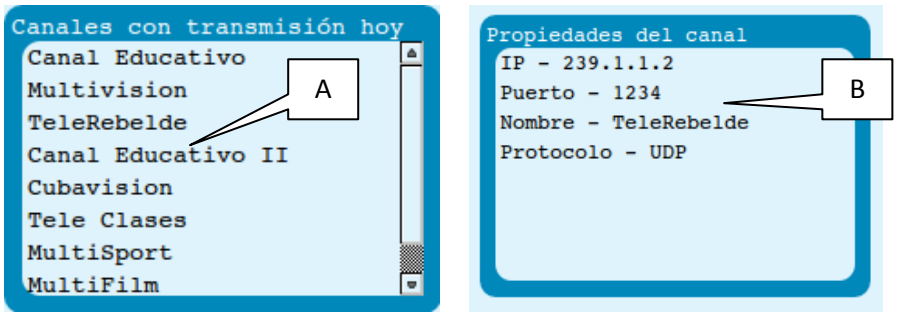



***Tabla 12: Descripción del Caso de Uso Visualizar Datos del Canal.***

### *3.5.3.7. Descripción del Caso de Uso Actualizar Transmisión.*

| | |
|---|---|
| **Caso de Uso:** | Actualizar Transmisión. |
| **Actores:** | Sistema |
| **Resumen:** | El caso de uso se inicializa cuando el sistema actualiza el flujo de los canales y la programación del día así como los datos de dichos canales, finaliza cuando termina la actualización completa del sistema. |
| **Precondiciones:** | Tienen que existir una conexión con la base de datos. |

| | |
|---|---|
| **Referencias** | R.7 |
| **Prioridad** | Critico |

**Flujo Normal de Eventos**

| Acción del Actor | Respuesta del Sistema |
|---|---|
| 1. Se inicializa el caso de uso cuando el sistema detecta de que hay algún cambio en la base de datos, de acuerdo a la programación que se tiene planificada para el día.<br><br>3. Finaliza el caso de uso cuando concluye la actualización del sistema (B). | 2. Al concluir uno de los canales que se están trasmitiendo, este será eliminado de la programación del día (A). |

**Flujos Alternos**

| | Respuesta del Sistema |
|---|---|
| | 2. Al iniciar un canal nuevo en la transmisión es agregado a la programación y transmisión del día (A). |

**Prototipo de Interfaz**


*Tabla 13: Descripción del Caso de Uso Actualizar Transmisión.*

### *3.5.3.8. Descripción del Caso de Uso Notificar Error en la Transmisión.*

| | |
|---|---|
| **Caso de Uso:** | Notificar Error en la Transmisión |
| **Actores:** | Sistema |

| | |
|---|---|
| **Resumen:** | Se inicializa el caso de uso cuando el sistema detecta un error en la transmisión y envía un aviso al transmisor, finalizando cuando envía el mensaje al transmisor. |
| **Precondiciones:** | Tiene que existir conexión con la base de datos.<br>Tiene que existir media en transmisión. |
| **Referencias** | R.8 |
| **Prioridad** | Critico |

**Flujo Normal de Eventos**

| Acción del Actor | Respuesta del Sistema |
|---|---|
| 1. Se inicializa el caso de uso cuando el sistema detecta error en la transmisión. | 2. Muestra una señal al transmisor de que existe un error (A).<br>3. El sistema le envía una notificación, informando de que hay un error en los canales que se están transmitiendo, especificando que canal tiene problemas además del tipo de error que presentó el mismo.<br>4. Finaliza el caso de uso cuando el sistema muestra de que existe un error en la transmisión, bordeando de color rojo la media que contiene el error. |

**Prototipo de Interfaz**

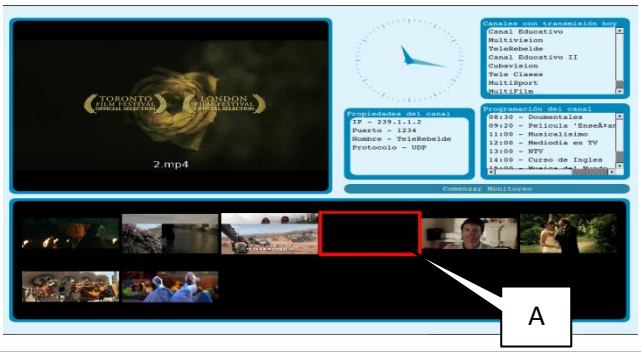



*Tabla 14: Descripción del Caso de Uso Notificar Error en la Transmisión*

### *3.5.3.9. Descripción del Caso de Uso Seleccionar Configuración de Entorno de Trabajo.*

| | |
|---|---|
| **Caso de Uso:** | Seleccionar Configuración de Entorno de Trabajo |
| **Actores:** | Transmisor |

| | |
|---|---|
| **Resumen:** | El caso de uso se inicia cuando el actor selecciona la opción referente al entorno de trabajo más conveniente, finalizando este cuando elige su entorno de trabajo. |
| **Precondiciones:** | El transmisor debe estar registrado para usar el sistema. |
| **Referencias** | R.9 |
| **Prioridad** | Critico |

**Flujo Normal de Eventos**

| Acción del Actor | Respuesta del Sistema |
|---|---|
| 1. El caso de uso se inicializa cuando el transmisor inicia la aplicación. | 2. El sistema muestra una pequeña interfaz dándole la posibilidad al transmisor de escoger el entorno de trabajo. |
| 3. Marca la opción más conveniente para el trabajo de monitoreo (A). | |
| 4. Le da clic al botón para abrir la ventana correspondiente al entorno de trabajo seleccionado (B). | 5. Finaliza el caso de uso en cuanto se muestra el entorno de trabajo deseado. |

**Prototipo de Interfaz**


*Tabla 15: Descripción del Caso de Uso Seleccionar Configuración de Entorno de Trabajo*

### *3.5.3.10. Descripción del Caso de Uso Visualizar Reporte de Errores.*

| | |
|---|---|
| **Caso de Uso:** | Visualizar Reporte Errores |
| **Actores:** | Sistema |

| | |
|---|---|
| **Resumen:** | Se inicializa el caso de uso cuando el transmisor desea ver los reportes de los errores que a presentado el sistema, finalizando en el instante en que son enviados de manera satisfactoria. |
| **Precondiciones:** | Tiene que existir alguna media en trasmisión. |
| **Referencias** | R.10 |
| **Prioridad** | Critico |

| **Flujo Normal de Eventos** | |
|---|---|
| **Acción del Actor** | **Respuesta del Sistema** |
| 1. Se inicializa el caso de uso cuando se desea ver los reportes de los errores del sistema y se va al menú para verlos(A). | 2. Se abre una ventana la cual muestra un listado de los reportes.<br><br>3. Finaliza el caso de uso cuando se cierra la ventana que muestra la lista y la grafica de los reportes. |
| **Flujos Alternos** | |
| | **Respuesta del Sistema** |
| | |

**Prototipo de Interfaz**


***Tabla 16: Descripción del Caso de Uso Generar Reporte de Errores***

## *3.6. Diagrama de Clases del Análisis.*

A continuación se realizarán los diagramas de clases del análisis. El objetivo del modelo de análisis es obtener una visión del sistema, de modo que solo se interesa por los requisitos funcionales. Es importante su desarrollo pues se garantiza la transición más fácil al modelo de diseño, además en caso de realizar algún cambio en las metodologías de desarrollo, habrá un modelo de análisis que respaldará el cambio

y garantizará que el mismo no sea brusco.

Para lograr una mejor organización del documento solo se mostrará a continuación la representación de los diagramas de análisis correspondientes a los siguientes casos de usos: Autenticar Usuario, Visualizar Programación del Día y por último el caso de uso Visualizar Media Seleccionada. Los restantes diagramas se podrán encontrar en los anexos al documento.

### *3.6.1. Autenticar Usuario.*

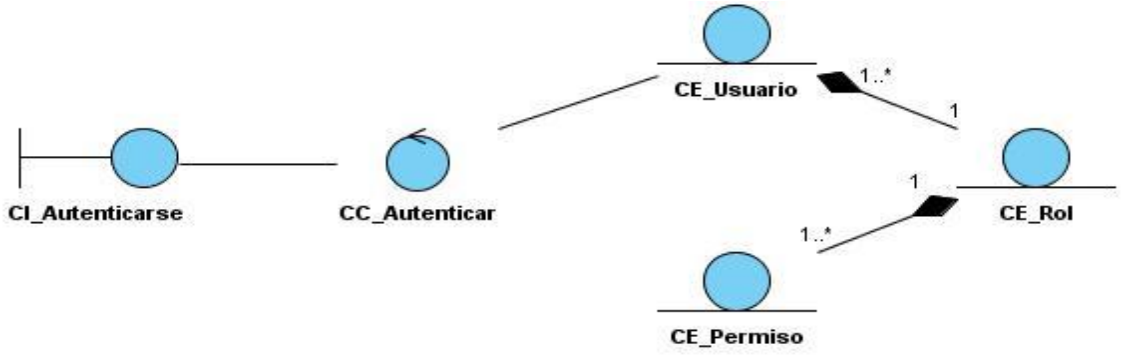



***Figura. 7: Diagrama de Clases del Análisis Autenticar Usuario***

### *3.6.2. Visualizar Programación del día.*

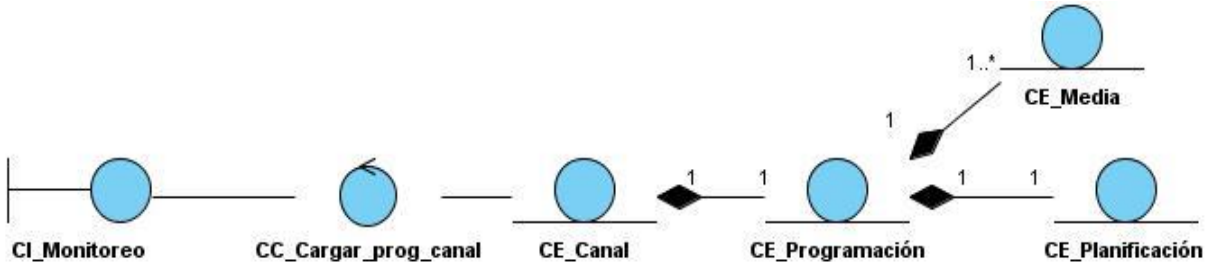



***Figura. 8: Diagrama de Clases del Análisis Visualizar Programación del Día.***

### *3.6.3. Visualizar Media Seleccionada.*

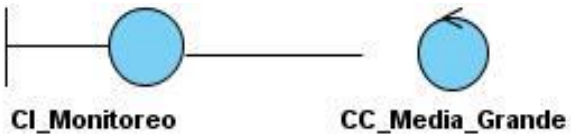



***Figura. 9: Diagrama de Clases del Análisis Visualizar Media Seleccionada.***

## *3.7. Conclusiones.*

Una vez concluido el presente capítulo quedó estructurada la propuesta de solución para la posterior implementación de la herramienta de monitoreo que se desea realizar, obteniéndose a partir de un análisis de las funcionalidades que debe poseer el sistema, las cuales se representaron mediante un Diagrama de Casos de Uso, describiéndose paso a paso todas las acciones de los actores del sistema con los casos de usos que interactúa.

Teniendo en cuenta todo lo abordado en este capítulo se da paso a construir el sistema. Poniendo en práctica el cumplimiento de los requisitos tanto funcionales como no funcionales que garantizan las condiciones y capacidades que debe generar el sistema.

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

## *4. Introducción.*

Durante el desarrollo de este capítulo se dará paso a la construcción de la solución propuesta del sistema. Posteriormente se realizará el diseño de la aplicación a través de diagramas de clases. Además se reflejarán los estándares de la interfaz y se realizará un modelo de implementación mediante la elaboración de los diagramas de componentes y el diagrama de despliegue.

## *4.1. Arquitectura.*

Se puede denominar arquitectura del software a la(s) estructura(s) del sistema, que incluyen los componentes del software, las propiedades visibles externamente de esos componentes y las relaciones entre ellos. La arquitectura permite la comunicación e interrelación entre los componentes del software. Es decir, constituye un modelo comprensible de cómo está estructurado el sistema y como trabajan e interactúan sus componentes.

Según Ian Somerville, la arquitectura de un sistema puede basarse en un modelo o estilo arquitectónico en particular. (28)

### *4.1.1. Estilo arquitectónico.*

Se ha asumido una aproximación al modelo de referencia arquitectónico en capas, definiéndose utilizar el estilo arquitectónico Tres Capas que se basa en el modelo de referencia especificado anteriormente. Se debe aclarar que esta arquitectura se corresponde con la definida para PTARTV. En este estilo se definen tres capas: (29)

- **Capa de Interfaz de Usuario o Presentación**: Reúne todos los aspectos del software que tiene que ver con las interfaces y la interacción con los diferentes tipos de usuarios. Estos aspectos típicamente incluyen el manejo y aspecto de las ventanas, el formato de los reportes, menú y gráficos en general. Representa la capa que interactúa con el usuario, también se le puede llamar interfaz de usuario.
- **Capa de Negocio**: Esta capa reúne todos los aspectos del software que tienen que automatizar o apoyan los procesos de negocio que llevan a cabo los usuarios. Estos aspectos típicamente incluyen las tareas que forman parte de

los procesos, las reglas y restricciones que aplican. Esta capa también recibe el nombre de la capa de la Lógica de la Aplicación. Es la capa donde se almacenan el código implementado, además de ser la capa encargada de transportar de manera bidireccional la información del gestor de base de dato, respondiendo a las solicitudes de los usuarios y Automatizando de esta manera los procesos de negocios que se llevan a cabo.

- **Capa de Acceso a Datos**: La capa de Acceso a Datos es el puente entre la capa de Lógica de Negocio y el Sistema de Base de Datos. Encapsula la lógica de acceso a datos. Aquí se encuentran componentes que hacen transparente el acceso a la base de datos. Este es el lugar idóneo para implementar los objetos de acceso a datos, permitiendo los mismos ingresar, obtener, actualizar y eliminar información del Sistema de Bases de Datos. Esta capa es la encargada de mantener el enlace entre la lógica de negocio y el gestor de base de datos, almacenando la información mediante el gestor utilizado.

### *4.1.2. Patrones de diseño.*

Los patrones están basados en la experiencia acumulada por los desarrolladores, con el objetivo de describir la solución más factible a problemas comunes, lo cual ayudará a no cometer errores consumados con anterioridad. Un patrón de diseño no es más que una solución estándar definida para cualquier problema de programación. Es una técnica para flexibilizar el código haciéndolo satisfacer criterios. Describe la conexión entre los componentes de un programa. Además es una manera práctica de definir aspectos de la organización de un sistema.

Un patrón de diseño es: (30)

- Una solución estándar para un problema común de programación.
- Una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios.
- Un proyecto o estructura de implementación que logra una finalidad determinada.
- Un lenguaje de programación de alto nivel.
- Una manera más práctica de describir ciertos aspectos de la organización de un programa.
- Conexiones entre componentes de programas.
- La forma de un diagrama de objeto o de un modelo de objeto.

Seguidamente se especifican los patrones de diseño utilizados que expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que se debe construir el software.

Para asignar la responsabilidad a un objeto determinado se sugiere el uso de los patrones de diseño GRASP, entre los que destacan: (31)

- **Experto**: La clase que cuenta con la información necesaria para cumplir una tarea debe ser la responsable de ejecutar la misma, proporcionando que los objetos exploten su propia información para cumplir con sus funcionalidades, lo cual conservará el encapsulamiento.
- **Creador**: Se ocupa de la creación de instancias en las clases, su tarea consiste en hallar un creador el cual se conectará con el objeto generado.
- **Alta cohesión**: Se ocupa de que las clases del diseño realicen las funcionalidades necesarias para cumplir con las tareas que tienen definidas. Ante una tarea de gran magnitud los objetos se comparten el trabajo colaborando entre ellos, evitando así la existencia de clases saturadas de métodos redundantes.
- **Bajo acoplamiento**: Tiene como objetivo establecer una escasa dependencia entre las clases, reduciendo el impacto de posibles cambios en el sistema.
- **Controlador**: Define quién deberá encargarse de atender un evento del sistema. Se debe diseñar de forma tal que no posean demasiada responsabilidad, delegando esta en otras clases mientras el controlador coordina la actividad.

## *4.2. Diagrama de Clases de Diseño.*

Los diagramas de clases de diseño constituyen un elemento fundamental en la concepción de la aplicación que se está proponiendo. Estos diagramas de clases de diseño sirven de guía a los desarrolladores para la construcción del sistema que se desea implementar. Para lograr una mejor organización del documento solo se representan los diagramas de clases del diseño correspondientes a los casos de uso Autenticar Usuario, Visualizar Programación del Día y Visualizar Media Seleccionada. Los restantes se anexan al documento.

### *4.2.1. Diagrama de Clase de Diseño del Caso de Uso Autenticar Usuario.*

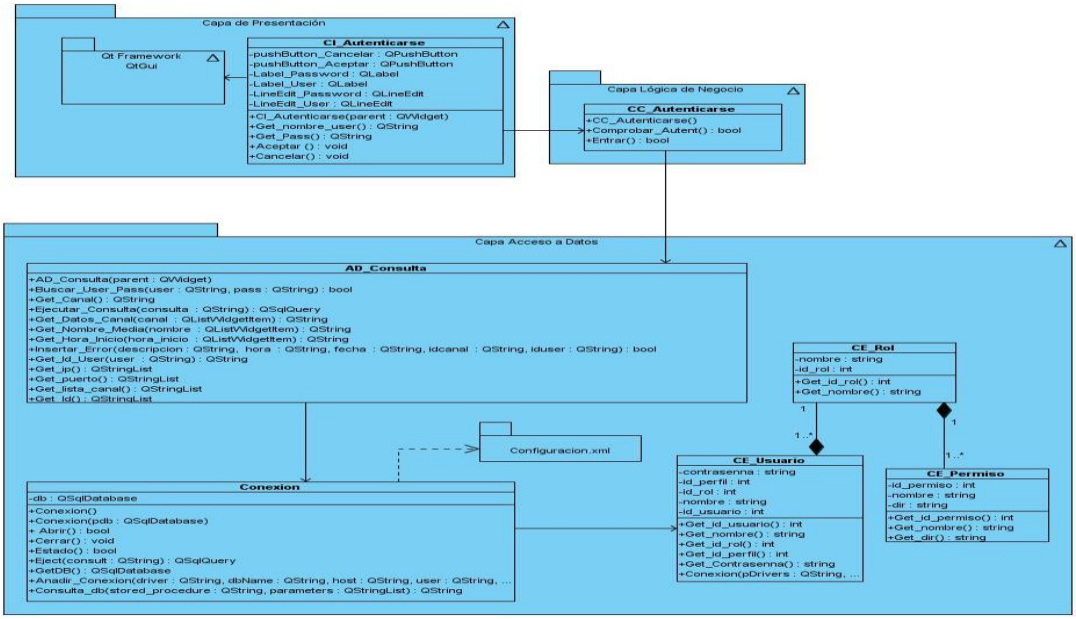



*Figura. 10: Diagrama de Clase de Diseño del Caso de Uso Autenticar Usuario.*

### *4.2.2. Diagrama de Clase de Diseño Visualizar Programación del Día.*

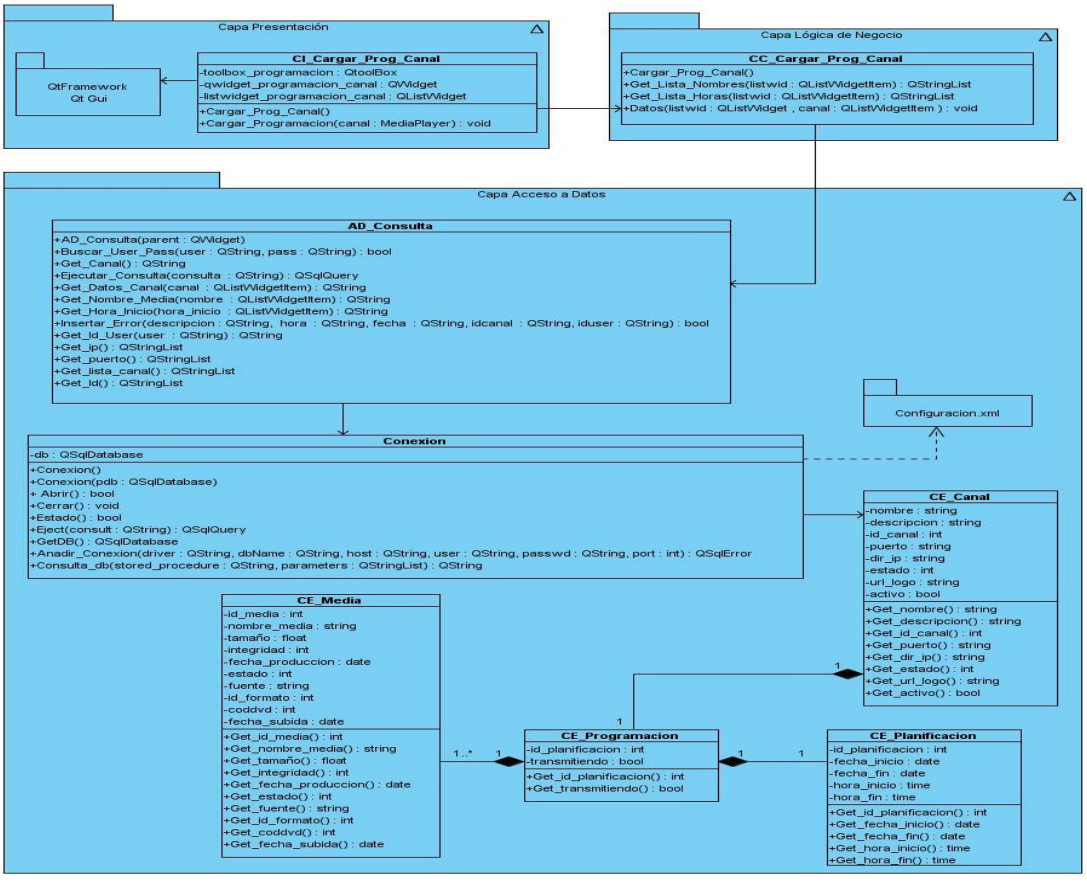



*Figura. 11: Diagrama de Clases de Diseño Visualizar Programación del Día.*

### *4.2.3. Diagrama de Clase del Diseño Visualizar Media Seleccionada.*

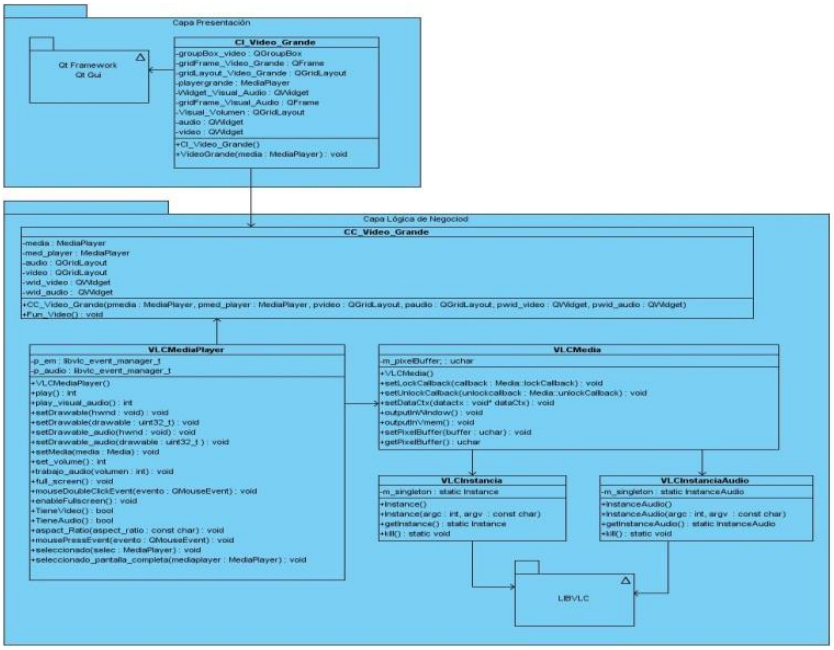



*Figura. 12: Diagrama de Clases del Diseño Visualizar Media Seleccionada.*

## *4.3. Modelo de Implementación.*

A continuación se representan los diagramas de componentes de los distintos casos de uso, los cuales presentan paquetes utilizados para agrupar los elementos físicos de un sistema, dígase componentes, interfaces, así como las relaciones entre ellos. Para lograr una mejor organización del documento solo se representan los diagramas correspondientes a los casos de uso Autenticar Usuario, Visualizar Programación del Día y Visualizar Media Seleccionada. Los Restantes se anexan al documento.

### *4.3.1. Diagrama de Componentes Autenticar Usuario.*

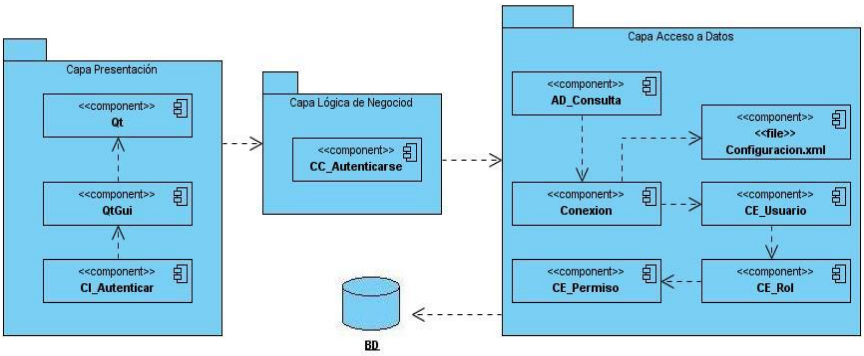



*Figura. 13: Diagrama de Componentes Autenticar Usuario.*

### *4.3.2. Diagrama de Componentes Visualizar Programación del Día.*

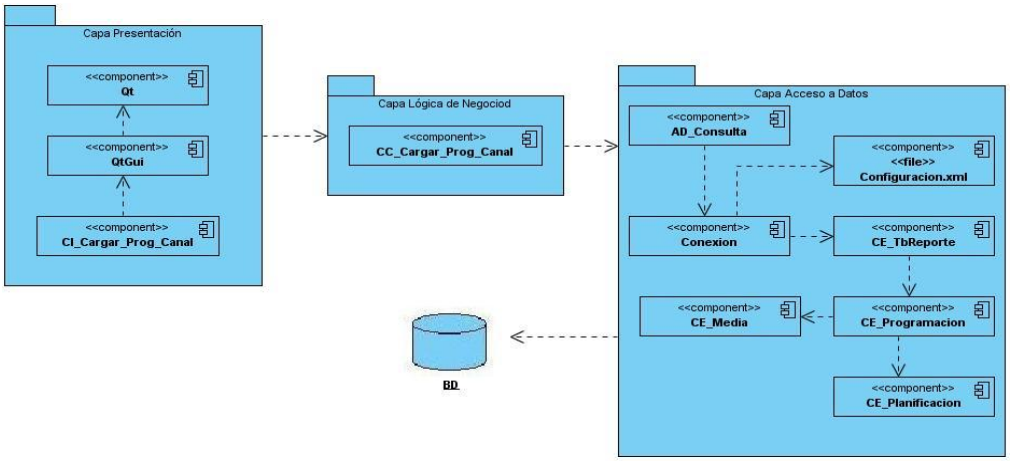



***Figura. 14: Diagrama de Componentes Visualizar Programación del Día.***

### *4.3.3. Diagrama de Componentes Visualizar Media Seleccionada.*

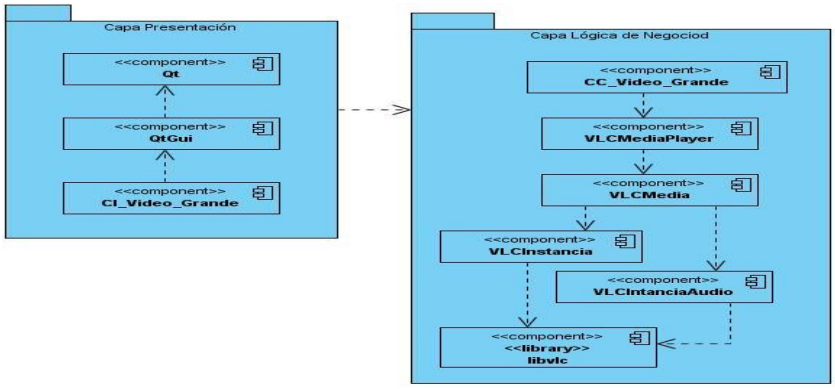



***Figura. 15: Diagrama de Componentes Visualizar Media Seleccionada.***

## *4.4. Modelo de Despliegue.*

En el Diagrama de Despliegue se muestran las relaciones físicas entre los componentes hardware y software en el sistema final. A continuación se muestran tres nodos, uno representa los ordenadores en los cuales estará el sistema. Los otros dos nodos constituyen los servidores donde se conectará la aplicación, para realizar su tarea fundamental que consiste en el monitoreo de las señales que se están transmitiendo.

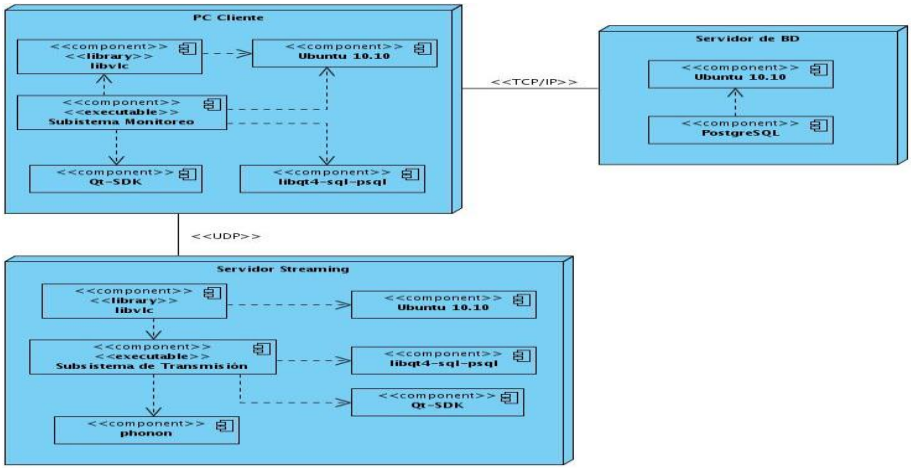



***Figura. 16: Modelo de Despliegue.***

## *4.5. Conclusiones.*

En el capítulo se ha planteado el modelo de construcción propuesto a través de los diagramas de clases y demás elementos necesarios para la implementación de la aplicación. Se explicó cómo está estructurada la aplicación físicamente, mediante los modelos de despliegue y de componentes, además de la definición del patrón arquitectónico en tres capas, los cuales permitieron dar paso a la implementación de la solución propuesta.

# *Capítulo 5: Validar la herramienta desarrollada.*

## *5. Introducción.*

La validación de una aplicación informática es un elemento esencial para garantizar la eficacia del software que se ha implementado y representa una revisión de las especificaciones del diseño y de la codificación de los elementos que componen la aplicación.

Existen varios tipos de pruebas, cada una de ellas con objetivos y habilidades bien definidas. En el este capítulo se describirán las pruebas funcionales además de las características esenciales de la herramienta diseñada e implementada.

## *5.2. Elementos del proceso de pruebas.*

Cuando se van a realizar las pruebas es necesario tener en cuenta los siguientes elementos: los niveles de prueba y tipos de pruebas a realizar. A continuación se explican de una manera más detallada dichos elementos.

### *5.2.1. Niveles de Pruebas.*

- **Prueba de desarrollador**: Indica los aspectos de diseño e implementación de las pruebas más adecuadas que debe llevar a cabo el equipo de desarrolladores, a diferencia de la prueba independiente. En la mayoría de los casos, la ejecución de la prueba se produce inicialmente con el grupo de pruebas de desarrollador que la diseñó e implementó; aunque es recomendable que los desarrolladores creen las pruebas de forma que estén disponibles para que las ejecuten grupos de pruebas independientes. (32)
- **Prueba independiente**: Indica el diseño y la implementación de la prueba realizada más adecuadamente por alguien ajeno al equipo de desarrolladores. Puede considerar esta distinción un súper conjunto, que incluye validación y verificación independientes. En la mayoría de los casos, la ejecución de la prueba se produce inicialmente con el grupo de pruebas independientes que la diseñó e implementó; aunque los verificadores independientes deberían crear sus pruebas de forma que estén disponibles para que las ejecuten los grupos de pruebas de desarrollador (33).
- **Prueba de unidad**: Se centra en la verificación de los elementos más pequeños del software que se puedan probar. Normalmente, las pruebas de

unidad se aplican a componentes representados en el modelo de implementación para verificar que se cubren los flujos de control y los flujos de datos y que funcionan como se esperaba. El implementador realiza la prueba de unidad mientras se desarrolla la unidad. Los detalles de la prueba de unidad se describen en la disciplina de implementación (33).

- **Prueba de integración**: Las pruebas de integración se realizan para garantizar que los componentes del modelo de implementación funcionan correctamente cuando se combinan para ejecutar un guión de uso. El destino de la prueba es un paquete o un conjunto de paquetes del modelo de implementación. A menudo, los paquetes que se combinan proceden de diferentes empresas de desarrollo. Las pruebas de integración exponen el estado incompleto o los errores de las especificaciones de la interfaz del paquete (33).
- **Prueba del sistema**: Normalmente, la prueba del sistema se realiza cuando el software funciona en su totalidad. Un ciclo vital repetitivo permite que las pruebas del sistema se realicen mucho antes, en cuanto se hayan implementado subconjuntos bien formados del comportamiento de guiones de uso. Normalmente, el destino son los elementos en funcionamiento de extremo a extremo del sistema (33).
- **Prueba de aceptación**: La prueba de aceptación del usuario es la última acción de prueba antes de desplegar el software. El objetivo de la prueba de aceptación es comprobar si el software está preparado y lo pueden utilizar los usuarios para realizar las funciones y tareas para las que se diseñó (33).

Luego de haber analizado los niveles de prueba por lo cuales debe transitar una aplicación informática, se deben conocer los tipos de pruebas que se aplican en cada uno de los niveles antes descritos.

### *5.2.2. Tipos de Pruebas.*

#### *5.2.2.1. Pruebas de Funcionalidad:*

- **Prueba de función**: pruebas que se centran en la validación de funciones del destino de la prueba, proporcionan los guiones de uso, los métodos y los servicios necesarios. Esta prueba se implementa y se ejecuta en diferentes destinos de la prueba, incluidas las unidades, las unidades integradas, las aplicaciones y los sistemas. (33)
- **Prueba de seguridad**: pruebas que se centran en garantizar que los datos del

destino de la prueba (o sistemas) sólo son accesibles para los actores a los que se dirigen. Esta prueba se implementa y ejecuta en varios destinos de la prueba. (33)

- **Prueba de volumen**: pruebas que se centran en la verificación de la capacidad del destino de la prueba para manejar grandes cantidades de datos, ya sean de entrada y salida o residentes, en la base de datos. La prueba de volumen incluye estrategias de prueba como la creación de consultas que devolverán el contenido completo de la base de datos, o que tendrán tantas restricciones que no devolverán ningún dato, o en las que la entrada de datos tiene la cantidad máxima de datos para cada campo. (33)

### *5.2.2.2. Pruebas de Fiabilidad*:

- **Prueba de integridad**: pruebas que se centran en la evaluación de la fuerza del destino de la prueba (resistencia a los errores) y la conformidad técnica del lenguaje, la sintaxis y la utilización de recursos. Esta prueba se implementa y se ejecuta en diferentes destinos de la prueba, incluidas las unidades y las unidades integradas. (33)
- **Prueba de estructura***:* pruebas que se centran en la evaluación de la adherencia del destino de la prueba a su diseño y formación. Normalmente, esta prueba se realiza en aplicaciones habilitadas para web y garantiza que todos los enlaces están conectados, se muestra el contenido adecuado y no hay ningún contenido huérfano. (33)
- **Prueba de tensión***:* se trata de un tipo de prueba de fiabilidad que se centra en la evaluación de cómo responde el sistema en circunstancias anormales. Las tensiones del sistema pueden ser cargas de trabajo extremas, memoria insuficiente, servicios y hardware no disponible o recursos compartidos limitados. Estas pruebas suelen realizarse para saber mejor cómo y en qué áreas fallará el sistema, de forma que se puedan planificar y presupuestar los planes de contingencia y el mantenimiento de las actualizaciones con bastante antelación. (33)
- **Prueba de puntos de referencia**: se trata de un tipo de prueba de rendimiento que compara el rendimiento de un destino de la prueba nuevo o desconocido con una referencia conocida, carga de trabajo y sistema. (33)
- **Prueba de contienda**: pruebas que se centran en la validación de la capacidad del destino de la prueba para manejar de forma aceptable varias demandas del actor en el mismo recurso (registros de datos, memoria, etc.). (33)

### *5.2.2.3. Rendimiento:*

- **Prueba de carga**: se trata de un tipo de prueba de rendimiento que se utiliza para validar y evaluar la aceptabilidad de los límites operativos de un sistema bajo cargas de trabajo variables, mientras el sistema que se está probando permanece igual. En algunas variantes, la carga de trabajo permanece igual y se modifica la configuración del sistema que se está probando. Las medidas suelen tomarse en función del rendimiento de la carga de trabajo y el tiempo de respuesta de las transacciones en línea. Las variaciones de la carga de trabajo suelen incluir la emulación del pico y el promedio de cargas de trabajo que se producen dentro de la tolerancia operativa normal. (33)
- **Perfil de rendimiento**: se trata de una prueba en la que se controla el perfil de tiempo del destino de la prueba, incluidos el flujo de la ejecución, el acceso de datos, las llamadas del sistema y de funciones para identificar y tratar los cuellos de botella de rendimiento y los procesos ineficaces. (33)
- **Prueba de configuración**: pruebas que se centran en garantizar que las funciones del destino de la prueba son las adecuadas en diferentes configuraciones de hardware y software. Esta prueba también se puede implementar como una prueba de rendimiento del sistema. (33)

### *5.2.2.4. Capacidad de Soporte:*

- Prueba de instalación: pruebas que se centran en garantizar que el destino de la prueba se instala correctamente en diferentes configuraciones de hardware y software, y en condiciones diferentes (como, por ejemplo, espacio de disco insuficiente o interrupciones de la alimentación). Esta prueba se implementa y ejecuta en aplicaciones y sistemas. (33)

Luego de conocer los tipos de pruebas que deben aplicarse a un producto de software, se puede seleccionar cual es el que se debe desarrollar para comprobar la calidad de la solución de la herramienta desarrollada. A continuación se aborda acerca de las características e importancias de realizar las pruebas funcionales.

## *5.3. 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, es común que este tipo de pruebas sean desarrolladas por analistas de pruebas con apoyo de algunos usuarios finales, esta etapa suele ser la última etapa de pruebas y al dar conformidad sobre esta el paso siguiente es el pase a producción (34).

A este tipo de pruebas se les denomina también pruebas de comportamiento o pruebas de caja negra, ya que los analistas de pruebas, no enfocan su atención a como se generan las respuestas del sistema, básicamente el enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida, esto generalmente se define en los casos de prueba preparados antes del inicio de las pruebas (34).

Las pruebas de caja negra se centran en los requisitos funcionales de software, permitiendo obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Este tipo de prueba intenta encontrar errores de las siguientes categorías: (34)

- Función incorrecta o ausente.
- Errores de interfaz.
- Errores en estructura de datos.
- Errores de rendimiento.
- Errores de inicialización y terminación.

### *5.3.1. Diseño de pruebas funcionales.*

Con el objetivo de poder validar el correcto funcionamiento de la aplicación se realizaron las pruebas de caja negra, que se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

Para la prueba de caja negra de este sistema se selecciona el caso de uso Autenticar Usuario.

**Caso de Prueba: Usuario correcto y contraseña correcta.**

| | |
|---|---|
| **Caso de Prueba** | Usuario y Contraseña correcta. |
| **Caso de uso** | Autenticar Usuario. |
| **Caso de prueba** | Comprobar que cuando el usuario va a entrar al sistema con su usuario y contraseña correcta puede acceder. |
| **Entrada** | Se accederá al sistema con un usuario Administrador y una contraseña Administrador. |
| **Resultado** | Como el usuario que existe en la base de datos es administrador y la contraseña administrador, los datos entrados son válidos. |

***Tabla 17: Caso de Prueba Usuario y Contraseña correcta.***

**Caso de Prueba: Usuario Incorrecto y Contraseña correcta.**

| | |
|---|---|
| **Caso de Prueba** | Usuario incorrecto y Contraseña correcta. |
| **Caso de uso** | Autenticar Usuario. |
| **Caso de prueba** | Comprobar que cuando el usuario escriba su usuario y su contraseña no lo haga con un usuario incorrecto y contraseña correcta. |
| **Entrada** | Se accederá al sistema con un usuario admin09 y una contraseña Administrador. |
| **Resultado** | Como el usuario que existe en la base de datos es Administrador y la contraseña no es la introducida correctamente, los datos entrados no son válidos y el sistema muestra un mensaje pidiendo que se chequeen los datos. Permitiendo que nuevamente vuelva a introducir sus datos. |

***Tabla 18: Caso de Prueba Usuario incorrecto y Contraseña correcta.***

**Caso de Prueba: Usuario correcto y Contraseña incorrecta.**

| | |
|---|---|
| **Caso de Prueba** | Usuario correcto y Contraseña incorrecta. |
| **Caso de uso** | Autenticar Usuario. |
| **Caso de prueba** | Comprobar que cuando el usuario escriba su usuario y su contraseña no lo haga con un usuario correcto y contraseña incorrecta. |
| **Entrada** | Se accederá al sistema con un usuario Administrador y |

| | |
|---|---|
| | una contraseña admin09. |
| **Resultado** | Como la contraseña que existe en la base de datos es Administrador y los datos del usuario no están correctamente, los datos entrados no son válidos y el sistema muestra un mensaje pidiendo que se chequeen los datos. Permitiendo que nuevamente vuelva a introducir sus datos. |

***Tabla 19: Caso de Prueba Usuario correcto y Contraseña incorrecta.***

**Caso de Prueba: Usuario incorrecto y Contraseña incorrecta.**

| | |
|---|---|
| **Caso de Prueba** | Usuario incorrecto y contraseña incorrecta. |
| **Caso de uso** | Autenticar Usuario. |
| **Caso de prueba** | Comprobar que cuando el usuario va a entrar al sistema con su usuario incorrecto y contraseña incorrecta no puede acceder al sistema. |
| **Entrada** | Se accederá al sistema con un usuario Admin09 y una contraseña Admin09. |
| **Resultado** | Como el usuario que existe en la base de datos es Administrador y la contraseña Administrador, los datos entrados no son válidos, por lo que el sistema muestra un mensaje pidiendo que se chequeen los datos. Permitiendo que nuevamente vuelva a introducir sus datos. |

***Tabla 20: Caso de Prueba Usuario incorrecto y Contraseña incorrecta.***

## *5.4. Pruebas de Rendimiento.*

En la Ingeniería del Software, las pruebas de rendimiento son aquellas que son realizadas para determinar que tan rápido un sistema realiza una tarea bajo ciertas condiciones pre-planificadas de trabajo. Estas pruebas también son utilizadas para validar y verificar diferentes aspectos de la calidad de software, como por ejemplo, escalabilidad, fiabilidad y el buen uso de los recursos. Las pruebas de rendimiento constituyen un subconjunto de la Ingeniería de Pruebas, la cual se esfuerza en mejorar el rendimiento, basándose en el diseño y la arquitectura de un sistema, antes de la realización del proceso de codificación (35).

### *5.4.1. Diseño de pruebas de rendimiento.*

Para probar el rendimiento de la Herramienta de Monitoreo con las señales que se están trasmitiendo en la Plataforma de Transmisión Abierta para Radio y Televisión, se realizo el monitoreo del consumo de memoria y de CPU.

El proceso se ejecutó en dos estaciones de trabajo con características diferentes, lo que permitió establecer límites en cuanto a la cantidad de canales a monitorear respondiendo a un hardware específico. A continuación se muestran las características de ambas estaciones, además de tablas que representan los niveles de consumo de memoria y de la CPU de la herramienta durante su funcionamiento:

**Estación de trabajo 1.**

- Sistema Operativo:
  - Ubuntu Release 10.10
  - Kernel Linux 2.6.32-21 generic.
  - GNOME 2.30.0.
- Hardware:
  - Memoria: 984 MiB.
  - Procesador 0: Intel(R) Core(TM)2 Duo CPU E4500 @ 2.20GHz.
  - Procesador 1: Intel(R) Core(TM)2 Duo CPU E4500 @ 2.20GHz.
- Estado del sistema:
  - Espacio libre en disco: 5.4 GiB.

**Resultados:**

| Cantidad de canales en transmisión. | Porciento consumido de la CPU. | Memoria consumida (MiB). |
|---|---|---|
| 4 | 45 | 200.6 |
| 8 | 80 | 447.5 |
| 12 | 98 | 710.0 |

***Tabla 21: Niveles de consumo de memoria y de la CPU para la estación de trabajo 1.***

**Estación de trabajo 2.**

- Sistema Operativo:
  - Ubuntu Release 10.10
  - Kernel Linux 2.6.35-22 generic.
  - GNOME 2.32.0.

- Hardware:
  - Memoria: 2.9 GiB.
  - Procesador 0: Intel(R) Core(TM) i3 CPU M 350 @ 2.27 GHz.
  - Procesador 1: Intel(R) Core(TM) i3 CPU M 350 @ 2.27 GHz.
  - Procesador 2: Intel(R) Core(TM) i3 CPU M 350 @ 2.27 GHz.
  - Procesador 3: Intel(R) Core(TM) i3 CPU M 350 @ 2.27 GHz.

- Estado del sistema:
  - Espacio libre en disco: 5.4 GiB.

**Resultados:**

| Cantidad de canales en transmisión. | Porciento consumido de la CPU. | Memoria consumida (MiB). |
|---|---|---|
| 4 | 20 | 115.3 |
| 8 | 40 | 180.7 |
| 12 | 50 | 238.9 |
| 16 | 55 | 324.0 |

***Tabla 22: Niveles de consumo de memoria y de la CPU para la estación de trabajo 2.***

## *5.5. Resultados de las Pruebas.*

En los epígrafes anteriores se detallaron las no conformidades encontradas durante las pruebas funcionales. Además se determinaron los límites de carga de la aplicación y el equipamiento necesario para su funcionamiento.

### *5.5.1. Resultados de las pruebas funcionales.*

Durante la fase de pruebas funcionales se encontraron elementos que afectaban el buen desarrollo de la aplicación. A continuación se detallan los elementos encontrados por los desarrolladores:

- Si no había conexión con la base de datos el sistema no mostraba ningún mensaje.
- Si no existían canales para la transmisión en ese día el sistema no mostraba ningún mensaje de que no habían canales en transmisión.

### *5.5.2. Resultados de las pruebas de rendimiento.*

Durante las pruebas de rendimiento se detectó, que el procesamiento correspondiente a la aplicación se sustenta sobre la capacidad operativa del microprocesador. Si el procesador puede atender mayor número de instrucciones por frecuencia de reloj se consumirá menor cantidad de memoria. Los niveles de consumo de memoria

aumentan enérgicamente cuando la carga de trabajo del procesador se acerca a su límite.

## *5.6. Conclusiones.*

La industria del desarrollo del software en el transcurso de los años es más exigente en cuanto a la calidad de los productos que se ofertan. Conocer los tipos de pruebas que pueden aplicársele a los sistemas y aplicaciones informáticas permitió seleccionar las más convenientes respecto a la solución propuesta.

A través de las pruebas seleccionadas se logró validar la actividad de los requisitos funcionales que se establecieron en la fase inicial del proceso de desarrollo del producto, verificando el correcto funcionamiento de la herramienta para el monitoreo de las señales digitales en la PTARTV.

Los resultados obtenidos durante la fase de pruebas de la herramienta representaron un elemento fundamental para corregir las deficiencias encontradas, para de esta manera dar por concluida la construcción de una herramienta que permitiese el monitoreo de diferentes señales que se estén transmitiendo en la PTARTV.

## *Conclusiones Generales.*

Con la caracterización de los procesos relacionados con el monitoreo de las señales que se trasmiten en la Dirección de Televisión Universitaria y en la Plataforma de Transmisión Abierta para Radio y Televisión, se logró la claridad necesaria para un mejor dominio del tema.

Se realizó una exhaustiva búsqueda bibliográfica acerca de las últimas tendencias y tecnologías que a nivel internacional se están utilizando en el mundo de la informática, para seleccionar aquellas que mejor respuesta darían al problema planteado: IDE Desarrollo (QT Creator), leguaje de programación (C++), como gestor de Base Datos (PostgreSQL) además se seleccionó a RUP como proceso de desarrollo de software a seguir para la construcción de la solución propuesta. Siguiendo este proceso, se diseñó e implementó la herramienta de monitoreo, con la cual se propone aumentar la eficiencia y confiabilidad del control de las señales televisivas.

Por tanto se arriba a las siguientes conclusiones:

- La aplicación favorece el control de la transmisión de las señales digitales de la Plataforma de transmisión abierta para radio y televisión así como en la DTU garantizando los niveles de seguridad de la información, mayor agilidad en los resultados, por lo que reduce los errores, además posee una interfaz sencilla fácil de utilizar.
- El objetivo general planteado durante la investigación fue cumplido, comprobándose la idea a defender como respuesta del problema a resolver.
- El uso de los métodos de investigación científica proporcionó un estudio detallado del objeto de estudio.
- Las tareas investigativas posibilitaron un mejor desarrollo del trabajo.

Para examinar la calidad y el correcto funcionamiento del sistema, se diseñaron y ejecutaron los casos de prueba, estos arrojaron resultados satisfactorios, demostrando el cumplimiento de los requisitos funcionales establecidos en la fase inicial del proceso de desarrollo del producto. Con el objetivo de examinar la calidad, se efectuaron pruebas de rendimiento que le permitieron al equipo determinar que hardware se necesitaba para desplegar la aplicación.

## *Recomendaciones.*

Se recomienda para próximas versiones realizar:

- Agregar ayuda al sistema.
- Agregar control de decibeles de audio, el cual permitirá el control de los canales de audio de los videos que de reproducen en la aplicación.
- Incluir una funcionalidad de poder gestionar y guardar la configuración de la aplicación de acuerdo a los gustos del usuario.

# *Bibliografía*

**1. Definicionde. [En línea] http://definicio.de/monitreo.**
**2.** Valle, Otto y Rivera, Otto. ***Monitoreo e Indicaciones.*** **Guatemala : Oficina Nacional en Guatemala.**
**3.** Española, Real Academia. **Diccionario de la lengua Española - Vigésima Segunda Edición. [En línea] Nivembre de 2010. http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=herramienta.**
**4. Tipo de Señales. [En línea] noviembre de 2010. http://www.linkses.com/articulos/471/Tipos_de_Señales.**
**5.** ***Transmisión de Datos.*** Compilacion y Pelliza, Armando Sergio. **2008.**
**6.** ***Sistema Digital y Sistema Analógico: conceptos, ventajas y ejemplos.*** Martín del Campo Becerra, Gustavo Daniel. **Guadalajara : s.n., 2008.**
**7.** TVLog. **TV Log. [En línea] 2 de febrero de 2008. http://www.tecnologiahechapalabra.com/tecnologia/comunicados/telecom/articulo.asp?i=2142.**
**8.** VideoSphere. **VideoSphere. [En línea] 2008. http://www.marchnetworks.com/Files/VS_VisualIntelligence_DS_SP_12-08.pdf.**
**9.** Omnicast. **Omnicast de Genetec. [En línea] 2010. http://www.instalacionprofesional.es/article.php?a=346.**
**10.** Ricardo Oswaldo, Francisco Xavier. ***Desarrollo de un sistema para determinar la ubicación geografica.*** **Quito : s.n., 2007.**
**11.** Departamento de Ciencias de la Computación, Universidad de Chile. **DCC. [En línea] 2010. http://www.dcc.uchile.cl/search/node/RUP.**
**12.** Calderón, Amaro, y otros. ***Metodologías Ágiles.*** **Trujillo Perú : s.n., 2007.**
**13.** Scribd. **Scribd. [En línea] 2010. http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE.**
**14.** Lezama, Jimenez y Carlos, Juan. ***Modelado de un BEAN de busqueda mediante UML.*** **2005.**
**15.** Alvarez, Sara. **Dasarrollo Web. [En línea] julio de 2007. http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html.**
**16.** Momjian, Bruce. ***PostgreSQL Introduction and concepts.*** **s.l. : Addison-Wesley, 2001.**
**17.** Cruz Chávez, Marco Antonio. **Universidad Autónoma del Estado de Morelos. [En línea] 2010. http://www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf.**
**18.** Schildt, Herbert. ***C++ Guía de Autoenseñanza.*** **España : s.n., 1995.**
**19.** Microsoft. **C++, Microsoft Search. [En línea] 2011. http://search.microsoft.com/results.aspx?form=MSHOME&mkt=es-ES&setlang=es-ES&q=c%23.**
**20.** Sarriegui, José María, y otros. **UNIVERSIDAD DE NAVARRA. [En línea] http://www.tayuda.com/ayudainf/index.htm.**
**21. Entorno Integral de Desarrollo. [En línea] 1998. http://ignetwork.net/showthread.php?15188-IDE-Entorno-integrado-de-desarrollo-%28Concepto-importante%29.**
**22.** Garrido, Salvador Alemany. **Introducción a Qt. 2011.**

**23.** NetBeans. **NetBeans. [En línea] http://netbeans.org/community/releases/69/.**
**24.** Ubuntu. **Guia de Ubuntu. [En línea] http://www.guia-ubuntu.org/index.php?title=NetBeans.**
**25.** Fowler, M. ***UML Destilled.*** **2000.**
**26.** Jacobson, Ivar, Booch, Grady y Rumbaugh, James. ***El Proceso Unificado de Desarrollo del Software.*** **Madrid : Addison Wesley, 2000. 84-7829-036-2.**
**27.** Scribd. **UML Diagrama de Actividad. [En línea] 2010. http://es.scribd.com/doc/2568098/UML-Diagramas-de-actividad.**
**28.** Sommerville, Ian. ***Igenieria de Software. 7ma Edición. Madrid.*** **s.l. : Editorial Pearson Education S.A., 2006.**
**29.** Teruel, Alejandro. **Arquitectura de capas. [En línea] 2011 de febrero de 10. http://www.ldc.usb.ve/~teruel/ci3715/clases/arqCapas.html..**
**30.** Berenguer, Abel Díaz y Vieito, Alberto Ramón Román. **Sistema de Administración y Configuración de la solución de Captura y Catalogación de Medias. Ciudad de La Habana : s.n., 2009.**
**31.** Larman, Craig. ***UML y Patrones.***
**32.** ***Ayuda de RUP.*** **s.l. : IBM Corp., 2006.**
**33.** ***Ayuda de RUP.*** **s.l. : IBM Corp., 2006.**
**34.** Oré B., Ing. Alexander. **CalidadySoftware.com.** ***FUNCTIONAL TESTING - PRUEBAS FUNCIONALES.*** **[En línea] 2009. [Citado el: 4 de Mayo de 2010.] http://www.calidadysoftware.com/testing/pruebas_funcionales.php.**
**35. Corporación Sybven-Integración Tecnológica. [En línea] 2011. [Citado el: 9 de mayo de 2011.] http://www.corporacionsybven.com/portal/index.php?option=com_content&view=article&id=246.**
**36.** Kent Beck, James Grenning, Robert C. Martin. ***Manifesto for Agile Software Development.*** **2001.**
**37.** Masip, David. **Desarrollo Web. [En línea] julio de 2002. http://www.desarrolloweb.com/articulos/840.php.**
**38.** Sunderic, By. **SQL Server. [En línea] http://doi.contentdirections.com/mr/mgh.jsp?doi=10.1036/0072133619.**
**39.** Marañón, Gonzalo Álvarez. **CSIC. [En línea] 1999. http://www.iec.csic.es.**
**40.** KDevelop. **KDevelop. [En línea] 2010. http://www.kdevelop.org/.**
**41.** Fiallo, Liseidis, Dayami, Hoyos y Bichara, Sierra. ***Propuesta de diseño de un Subsistema para la Consulta de Información referente a Concesionarios Mineros en la Oficina Nacional de Recursos Minerales.*** **2010.**

# *Glosario de términos.*

**Aplicación:** Es un programa informático creado para facilitar al usuario un determinado grupo de actividades.

**Multiplataforma:** Se utiliza este tipo de término para denominar a los programas, lenguajes de programación u otra clase de software que pueden brindar sus prestaciones funcionando sobre diversas combinaciones de hardware y software.

**Streaming:** Es un flujo de información digital que permite la reproducción de sonido o vídeo sin que sea necesario descargar previamente todo el archivo de recurso.

**Televisión:** Es un medio de comunicación que combina los mensajes mediante imágenes fijas y en movimiento con voz, música y efectos sonoros.

**Transmisión:** Es la emisión de una señal de radio o televisión desde una estación emisora hacia otra que actúa como receptora de la misma.

**Transmisión Abierta:** Es una transmisión de libre acceso, por la cual no es necesario pagar para consumirla.

**Video:** Es una tecnología de captura electrónica, tratamiento, almacenamiento, transmisión y reconstrucción de una sucesión de imágenes y sonidos que representan un conjunto de escenas en movimiento.